People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University of Abdelhamid Mehri – Constantine 2
Faculty of New Technologies of Information and Communication (NTIC)
Department of Fundamental Computing and its Applications (IFA)

MASTER'S THESIS

to obtain the diploma of Master degree in Computer Science

Option: Data Science and Artificial Intelligence

AI agents for real-time ECG interpretation

Realized by:
Guebli Ayoub Abdessami
Triai Radja

Under supervision of:
Dr.Talbi Hassiba
Dr.Bahri Mohamed redha

June 2025

بسم الله الرحمن الرحيم

Acknowledgments

We would like to express our deepest gratitude to **Dr. Talbi Hassiba** and **Dr. Bahri Mohamed Redha** for their guidance, expertise, and valuable feedback throughout the duration of this work. Their support played a crucial role in helping us develop this thesis both technically and academically.

We also extend our sincere appreciation to the members of the **thesis jury** for their time and for accepting to evaluate this work. Their insights and recommendations are greatly valued.

We are thankful to the faculty and staff of the **Department of Fundamental Computing and its Applications (IFA)** at **Constantine 2 University** for providing a supportive academic environment and the necessary resources to carry out this research.

Lastly, We are profoundly grateful to everyone who contributed, in any way, to the successful completion of this project — whether through encouragement, advice, or moral support.

Dedication

To our Parents, Families, and Friends,

We are deeply grateful for the unwavering support and guidance you have provided us.

Our parents and families — your love and sacrifices have shaped us into who we are today.

To all our friends — your friendship has added joy and adventure to our journey.

And to all our classmates and professors — thank you for being integral parts of our lives.

ملخص

تقترح هذه الدراسة نظامًا **ديناميكيًا** و **ذكيًا** لتحليل تخطيط القلب الكهربائي (ECG) يتميّز بـ **التفسير** و **الشرح** ، ويعمل في **الوقت الحقيقي** أو عبر **معالجة دفعات** من الإشارات، وذلك ضمن **بنية متعددة العوامل** (Multi-Agent System) تهدف إلى تحسين **القابلية للتوسعة** ، و **المرونة** ، و **فصل المهام** .

يبدأ النظام بـ **مرحلة المعالجة المسبقة** ، تليها عملية **تقسيم الموجات** باستخدام نموذج U-Net أحادي البعد مدرّب بـ **خسارة** Focal Loss مرجّحة، وقد تفوّق على نماذج بديلة مثل TCN و CNN-LSTM . ينتج عن التقسيم **قناع يُحدد موجات** P **و** QRS **و** T .

بعد ذلك، يتم استخراج **خصائص شكلية وإيقاعية** لكل نبضة قلب، مما يسمح بـ **تصنيفها على مستوى النبضة** إلى ست فئات (مثل النبض الطبيعي، الحصار الأيسر، الحصار الأيمن، نبض اصطناعي، بطيني، أو أخرى). تُستخدم هذه الخصائص لاحقًا لحساب **مؤشرات إحصائية عامة** (المتوسط، الانحراف المعياري، الحد الأدنى، الحد الأقصى، توزيع الفئات)، ويتم تصنيف الإشارة كاملة إلى **آعاديةْ** أو **آغير عاديةْ** . إذا كانت غير عادية، تُستخدم **نماذج متخصصة في الأمراض** لاكتشاف حالات مثل **بطء الجيب** ، **تسرّع الجيب** ، أو **اضطرابات النظم** باستخدام **مصنفات** Random Forest .

تم تطوير المعمارية باستخدام **وكلاء** SPADE عبر بروتوكول XMPP ، ودمجها في **واجهة خلفية** Django ، مع دعم **واجهات** REST و WebSocket للمراقبة الحية (مثل حالات العناية المركزة). يوفّر النظام **دقة عالية** و **قابلية للتفسير** تسمح للطبيب بالاطلاع على الإشارة والخصائص المؤثرة في القرار.

الكلمات المفتاحية: الذكاء الاصطناعي، تخطيط القلب الكهربائي، تعلم الآلة، التصنيف، التحليل في الوقت الفعلي، الأنظمة متعددة الوكلاء

Abstract

This study proposes a **dynamic** and **interpretable** ECG analysis system for **real-time** and **batch signal classification**, developed within a **multi-agent architecture**. The goal is to improve **modularity**, **scalability**, and **clinical explainability** using a clean design where each task is handled by an independent agent.

The system begins with **preprocessing** and **waveform segmentation** using a **1D U-Net** trained with **weighted Focal Loss**, outperforming alternatives like **TCN** and **CNN-LSTM**. Segmentation produces masks identifying **P, QRS, and T waves**. From these, **morphological** and **rhythm-based features** are extracted for each heartbeat, enabling **beat-level classification** into six types (e.g., Normal, LBBB, RBBB, Paced, Ventricular, Other). **Global features** (e.g., mean/variance of durations and wave amplitudes, class distributions) are then computed to classify the entire signal as **normal or abnormal**. If abnormal, **disease-specific models** detect conditions like **Sinus Bradycardia**, **Sinus Tachycardia**, or **Arrhythmia**, using **Random Forest classifiers**.

The architecture is built using **SPADE agents** over **XMPP**, embedded in a **Django backend** with **REST APIs** and **WebSockets** for live monitoring (e.g., in ICU scenarios). The system provides **high accuracy** and **interpretability**, allowing clinicians to view the underlying waveforms and extracted features that drive decisions.

Keywords: Artificial intelligence, ECG, machine learning, classification, real-time analysis, multi-agent systems

Résumé

Cette étude propose un système **intelligent**, **dynamique** et **interprétable** pour l'analyse des signaux **ECG**, capable de fonctionner **en temps réel** ou en **mode batch**. L'architecture repose sur un **système multi-agents (MAS)**, permettant une **modularité**, une **évolutivité** et une **séparation claire des tâches**.
Le pipeline commence par une phase de **prétraitement**, suivie d'une **segmentation des ondes P, QRS et T** à l'aide d'un modèle **U-Net 1D** entraîné avec une **fonction de perte Focal Loss pondérée**. Ce modèle a surpassé d'autres architectures telles que **TCN** et **CNN-LSTM**. Après la segmentation, des **caractéristiques morphologiques et rythmiques** sont extraites pour chaque battement cardiaque, permettant leur **classification** en six catégories (Normal, LBBB, RBBB, stimulé, ventriculaire et autres). Ensuite, des **statistiques globales** (moyenne, écart-type, min/max) sont calculées pour **classer le signal complet** comme **normal ou anormal**. En cas d'**anomalie**, des **modèles supplémentaires** détectent des pathologies spécifiques comme la **bradycardie sinusale**, la **tachycardie sinusale**, ou l'**arythmie**.
L'ensemble est orchestré par des **agents SPADE** via le **protocole XMPP**, intégré à une **plateforme Django** avec une **API REST** pour l'analyse différée et **WebSocket** pour la **surveillance en temps réel**. Le système offre une **précision élevée** et une **grande explicabilité** pour accompagner le **diagnostic médical**.

Mots clés : Intelligence artificielle, électrocardiogramme, apprentissage automatique, classification, analyse en temps réel, systèmes multi-agents

Table of Contents

List of Figures

List of Tables

List of Algorithms

General Introduction

Project Background

Cardiovascular diseases (CVDs) still rank as the leading cause of death, making early detection and monitoring of heart conditions essential[62]. Electrocardiography(ECG) is a non-invasive procedure of recording the electrical activities of the heart[57]. It plays the central role of diagnosis of many cardiac diseases like arrhythmias, myocardial infarction, and conduction abnormalities[49]. The ECG signal comprises several waveforms which indicate different phase in the cardiac cycle: P wave, QRS complex, and T wave[8]. The components must therefore carefully be analyzed in making clinical decisions.

ECG interpretation has traditionally been conducted by trained doctors of medicine or cardiologists manually. However, it is a repetitive and time consuming task and can be impacted by human fatigue, lack of experience and variations in interpretation from one expert to the other[49]. These disadvantages are particularly conspicuous in heavy workload, continuous recording, and urgent analysis.

In the last years, artificial intelligence (AI) has proved to be a valuable aid for the medical doctor, providing a new opportunity to automatize and enhance the diagnosis. For ECG analysis, the AI technology can also be used for detection of heart waveforms, feature extraction and diagnostic recommendations[20].

Problem

With the growth of healthcare systems, demand on automated ECG interpretation has become more and more urgent[49]. Manually performed by clinicians, it is time-consuming (speaking of the time that can absolutely not be done without), prone to fatigue and variable, and not scalable. It is critical to automate such analysis, to guarantee accurate, fast, and widely available assessment of cardiac exam, most importantly in situations where expertise is scarce or high-throughput is required.

Despite the wide use of ECG in clinical practice, several important challenges remain when it comes to automating its interpretation.

First, ECG signals obtained in real world conditions are noisy. Artifacts are introduced by movements, poor electrode contact, and signal distortion which make it difficult to precisely locate the relevant waves (P, QRS, T). If these waves are not measured accurately, any subsequent analysis or diagnosis is meaningless[54]. Hence, the preprocessing and wave segmentation must be robust and correct.

Second, most current ECG systems either perform segmentation or classification — not both. This separation creates a lack of continuity between low-level signal understanding and clinical reasoning[49]. In other words, these systems may detect a QRS complex or a T wave, but they do not always explain what it means medically (e.g., ST depression or prolonged QT interval). This reduces their usefulness in real diagnostic support.

Thirdly, even if automated systems are employed, they are generally perceived to have limited transparency and explainability. In critical care scenarios, clinicians must know how the system arrived at a particular decision. If an AI model makes a diagnosis without revealing the features it has learned, doctors may not be willing to trust it[20]. This is especially problematic in the field of cardiology, where mistaken readings can have tragic results.

Finally, real-time or long-term monitoring systems generate large volumes of data. Efficient feature extraction, interpretation, and data summarization become necessary to avoid overwhelming medical staff[54]. The ability to detect important features in real-time and summarize the results in a meaningful way is still a technical challenge.

Proposed Solutions

The objective of this project is to develop an intelligent ECG interpretation system using a modular, multi-agent architecture. Each agent is responsible for a distinct stage in the processing pipeline, ensuring clarity, flexibility, and maintainability. The system is designed to take raw ECG signals as input and handle the analysis step by step, from capturing the signal to producing a medical decision[20].

The core components of the system include:

- **Signal Acquisition and Preprocessing:** Reading the ECG files and extracting signals from them. Then apply preprocessing on this signals by bandpass filtering, smoothing, normalization, and resampling to ensure consistency and enhance quality.

- **Wave Segmentation:** A 1D U-NET model is used to detect and mark the key ECG components —P waves, QRS complexes and T-wave— from the preprocessed signal. The resulting masks identify accurate waves for further analysis[20].
- **Feature Extraction:** The system extracts the necessary clinical features from segmented waves, focusing on patterns that may indicate potential abnormalities. These features serve as the basis for downstream interpretation[49].
- **Beat Classification:** Using the extracted features, a Random Forest model classifies each individual heartbeat, identifying it as normal or specifying a type of abnormality.
- **Signal-Level Diagnosis:** final decision agent summarizes comprehensive signal features extracted from all beats. Using these full signal features, it classifies the overall signal and detects potential pathologies through Random Forest models[20].
- **Principal Agent and User Interface:** This agent manages inter-agent communication, coordinates outputs, and send it to User Interface, that will present the final analysis in a clear and accessible format through a user-friendly web interface[20].

The overall system aims to be scalable, interpretable, explanatory and optimal for various use cases, such as real -time patient monitoring, offline analysis and academic application. By structuring the pipeline in independent but cooperative agents, architecture promotes transparency, modularity and continuous improvement[20; 49].

Document Plan

This thesis is organized as follows:

- **Chapter 1** offers a wide state of the art review, which covers the physical background of ECG signals, current computational approaches for ECG analysis and challenges made.
- **Chapter 2** presents our main contribution. This gives details of the theoretical proposal of our intelligent ECG interpretation system, including the integration of agent-based architecture, model options and full structure. This chapter also describes experiments made to evaluate practical implementation of our system, datasets used, processing strategies and its performance.
- **General Conclusion** summarizes the major conclusions and boundaries of this task and underlines possible directions for future research and development.

This structure ensures a logical flow from background information and reference, through the detailed design and implementation of our proposed system, for a reflection on its importance and potential effect.

Chapter

1

State of the Art

Introduction

Electrocardiography (ECG) is a widely used, non-invasive technique for monitoring the electrical activity of the heart. It plays a crucial role in diagnosing various cardiac conditions, such as arrhythmias, heart attacks (myocardial infarctions), and conduction disorders [11; 61]. Despite its clinical importance, ECG interpretation remains a complex task—especially in busy hospitals or clinics, or in places where there are limited resources [28].

To address these challenges, more and more research has focused on automating ECG analysis using advanced signal processing, machine learning, and deep learning techniques [39; 43; 44]. These approaches aim to enhance the speed, accuracy, and consistency of ECG interpretation while reducing the amount of work doctors and nurses have to do. However, many existing systems still face challenges related to generalizability, noise robustness, transparency, and real-time performance [29; 35; 40].

This chapter provides an overview of how ECG analysis with AI is currently being done and how artificial intelligence (AI) is being used to support medical decision-making. It begins with a review of the physiological basis of ECG signals and their importance in medical diagnosis. Next, it explores the core components of computational ECG analysis—such as wave segmentation, morphological feature extraction, and classification techniques.

Special attention is given to the shift from traditional algorithms to deep learning-based models, as well as to recent efforts in improving model interpretability through explainable AI techniques [4; 60]. Finally, the chapter examines the emerging role of intelligent agents and multi-agent systems in healthcare, which are increasingly being used to create flexible, intelligent systems that adapt to different clinical needs [23; 51].

By analyzing both the strengths and limitations of existing methods, this review highlights the key technical and practical challenges that keep researchers working on new and better solutions.

1.1 Project Context and Area

1.1.1 Physiological Foundations of the Heart

a Heart Anatomy

The human heart is a muscular organ located slightly to the left of the center of the chest, between the lungs[58]. It is protected by a double-layered sac called the **pericardium**, which contains a fluid that reduces friction as the heart beats[53].

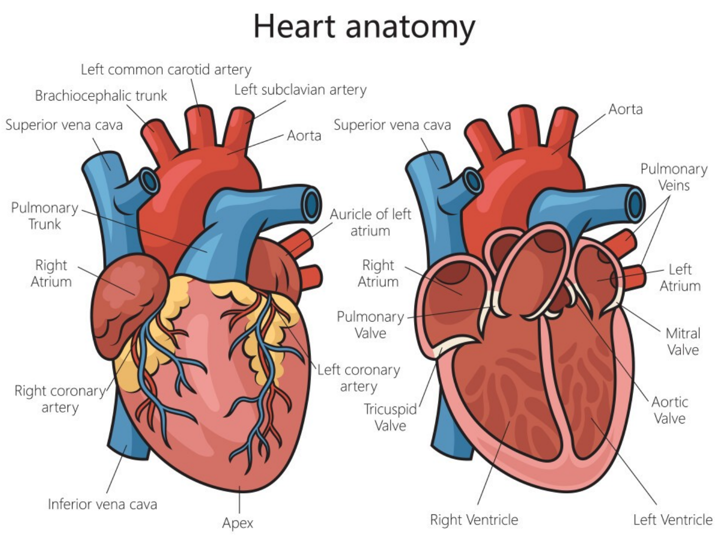


Figure 1.1: Anatomical diagram of the heart with labeled chambers, valves, and major vessels. Source: VectorStock [59].

The heart is divided into four chambers, and uses four valves to regulate blood flow. These are summarized in Table 1.1.

Table 1.1: **Heart Chambers and Valves**

Component	Function
Right Atrium	Receives oxygen-poor blood from the body via the superior and inferior vena cava.
Right Ventricle	Pumps this blood to the lungs through the pulmonary artery.
Left Atrium	Receives oxygen-rich blood from the lungs via the pulmonary veins.
Left Ventricle	Pumps oxygenated blood to the body through the aorta.
Tricuspid Valve	Located between the right atrium and right ventricle.
Pulmonary Valve	Located between the right ventricle and pulmonary artery.
Mitral Valve	Located between the left atrium and left ventricle.
Aortic Valve	Located between the left ventricle and aorta.

These chambers work in pairs — the right side of the heart manages circulation to the lungs, while the left side supplies the rest of the body[9].

Heart Wall and Circulation The wall of the heart has three layers:[53]

- **Endocardium** — the smooth inner lining.
- **Myocardium** — the thick, muscular middle layer responsible for pumping.
- **Epicardium** — the outer layer that protects the heart.

The heart receives its blood supply from the **coronary arteries**, which branch out from the aorta and deliver oxygen to the heart muscle itself. Without this blood flow, the myocardium would be damaged, which can result in a heart attack[56].

b Heart Electrophysiology

The heart's ability to beat in a regular, synchronized manner is controlled by its electrical conduction system. This system is composed of specialized cells and nodes that produce and transmit electrical impulses[8]. These impulses ensure the coordinated contraction of heart muscles, allowing efficient blood circulation throughout the body[8].

The Cardiac Conduction System The conduction system includes two main types of cells:

- **Conducting cells**, which carry the electrical signals across the heart.

- **Muscle cells**, which respond to these signals by contracting and enabling blood flow.

These cells work together to maintain a consistent rhythm and allow the heart to adjust to the body's changing demands — such as increased heart rate during physical activity or slowing down during rest[8].

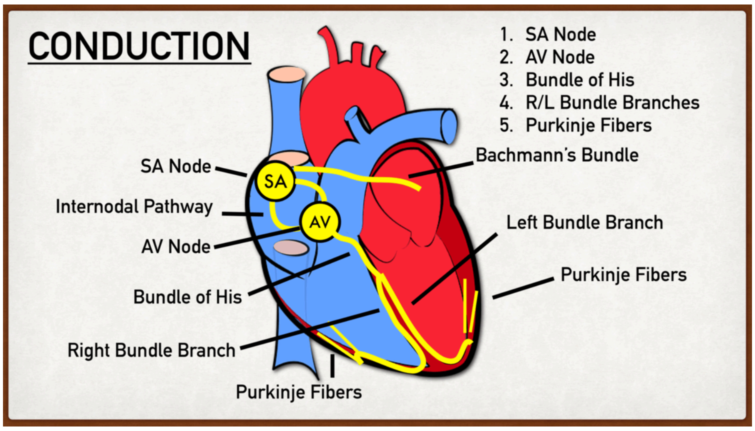


Figure 1.2: Electrical conduction pathway of the heart. Source: EZmed Learning [15].

How the Conduction Pathway Works Each heartbeat starts when the **sinoatrial (SA) node**, located in the right atrium, generates an electrical signal. This signal spreads across both atria, causing them to contract and push blood into the ventricles. Next, the signal reaches the **atrioventricular (AV) node**, where it is briefly delayed. The signal then travels through the **bundle of His** and down into the **Purkinje fibers**, which spread throughout the ventricles, triggering their contraction[8].

Physiological Importance The heart's electrical system repeats this process continuously to keep the heartbeat steady. It ensures blood is pumped correctly with each beat. If there is any issue in this system — like a delay or a wrong signal — the heart may beat too fast, too slow, or unevenly. These irregular rhythms are called arrhythmias[8].

At the core of this electrical activity are *action potentials* — tiny changes in the electric charge of heart cells. These changes trigger the contractions that produce the waves seen in an ECG, making them essential for both heart function and ECG analysis[49].

c Electrocardiography

Electrocardiography (ECG or EKG) is a simple, fast, and non-invasive medical test used to measure the electrical activity of the heart[57]. It plays an essential role in cardiology, helping doctors diagnose and monitor many cardiac problems by providing a visual trace of the heart's electrical signals[49].

How Electrocardiography Works During each heartbeat, the heart produces small electrical impulses that spread through the body. These weak signals can be picked up on the surface of the skin using electrodes[57]. In a standard ECG, electrodes are placed on the chest, arms, and legs. These electrodes detect the direction and strength of the electrical activity and send this information to a machine, which draws a graph — the ECG trace[57].

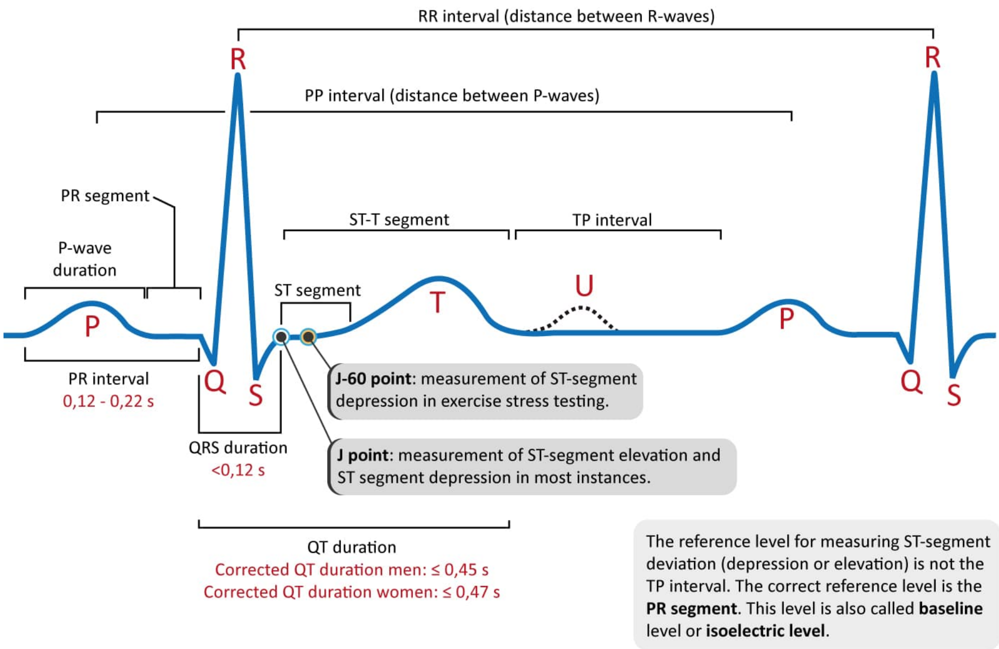


Figure 1.3: Labeled ECG waveform showing the P wave, QRS complex, T wave, and key intervals. Source: ECGWaves.com [13].

Understanding the ECG Waveform The ECG trace is made up of waves that repeat with each heartbeat. Each wave represents a specific electrical phase of the heart cycle[8]:

- **P wave**: This small bump shows the electrical activity that causes the atria (upper chambers) to contract and move blood into the ventricles (lower chambers).
- **QRS complex**: This is a sharp spike on the graph. It represents the electrical signal that causes the ventricles (lower chambers) to contract and pump blood to the lungs and body. It is the strongest part of the signal.
- **T wave**: This wider and rounder wave comes after the QRS complex. It reflects the recovery or "resetting" of the ventricles as they prepare for the next heartbeat.

The shape, size, and timing of these waves are essential indicators of the heart's electrical and muscular health[49].

1.1.2 Clinical Importance of ECG

The electrocardiogram (ECG) is one of the most common and important diagnostic tests in cardiology. It provides a non-invasive, real-time record of the heart's electrical activity and is used to detect a wide range of conditions, such as arrhythmias, heart attacks (myocardial infarctions), and conduction problems [61]. Because it is fast, affordable, and widely available, the ECG is often the first test performed when a patient reports symptoms like chest pain, dizziness, shortness of breath, or palpitations.

In some cases, doctors may use portable ECG devices such as Holter monitors to record heart activity over 24 hours or more, especially when symptoms are irregular or don't appear during a standard exam [61]. While the ECG is clinically essential, interpreting the signals can be challenging. Manual analysis depends on expert knowledge and may differ from one doctor to another. It is also prone to errors, particularly during long recordings or when the signal is noisy.

These difficulties have led to increasing interest in using computational tools and intelligent systems to make ECG interpretation faster, more consistent, and scalable in everyday clinical practice.

1.2 Related Works

1.2.1 ECG Segmentation

a Introduction to ECG Segmentation

Electrocardiogram (ECG) segmentation is the process of identifying the key parts of each heartbeat signal — the P wave, QRS complex, and T wave. This step is essential because it helps doctors and systems interpret the ECG correctly. It also helps measure how long the waveforms last, the gaps between them, and any small changes that might point to heart issues [22; 39; 66].

Earlier techniques such as wavelet transforms and dynamic time warping were commonly used for this task. While they worked reasonably well, they often struggled with noisy signals and variations between patients. In recent years, these older methods have been mostly replaced by deep learning models like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and U-Net architectures [5; 39].

These newer models are better suited for ECG segmentation because they can learn patterns directly from raw signals — even when the signals are messy or inconsistent. CNNs are good at capturing the shape of the waves, RNNs help recognize how the signal changes over time, and U-Nets combine both detailed and overall information to improve segmentation accuracy.

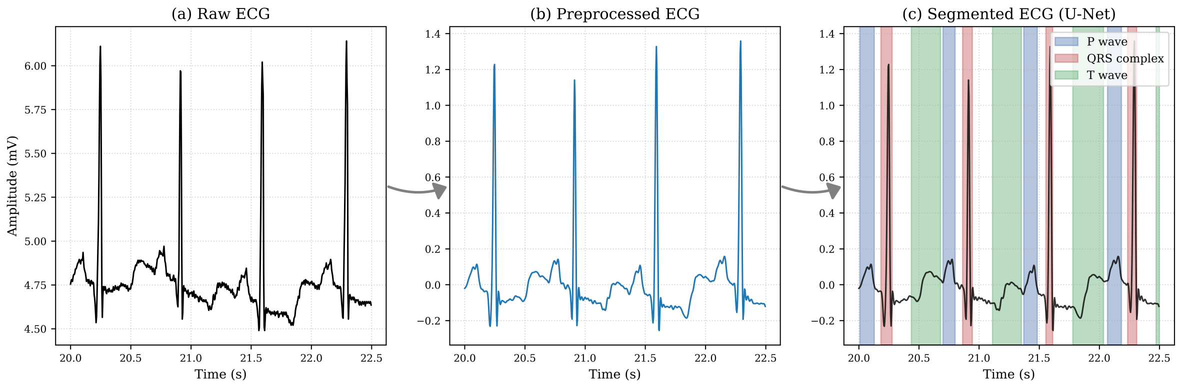


Figure 1.4: ECG segmentation pipeline with preprocessing and U-Net output.

b Preprocessing

Preprocessing improves ECG signal quality by cleaning out different types of noise that can make it hard to detect heart waves accurately

Table 1.2: Common ECG Noise Types and Removal Methods

Noise Type	Description and Removal
Baseline Wander	Low-frequency drift; high-pass filtering [29].
Powerline Interference	50/60 Hz noise; removed via notch filter [54].
EMG Noise	Muscle activity noise; filtered adaptively [54].
Electrode Motion	Movement artifacts; minimized by stabilization [29].

To prepare the signals for analysis, additional steps like bandpass filtering, normalization, resampling, and smoothing are often used. These help reduce noise and make the ECG more consistent across different patients [29; 54]

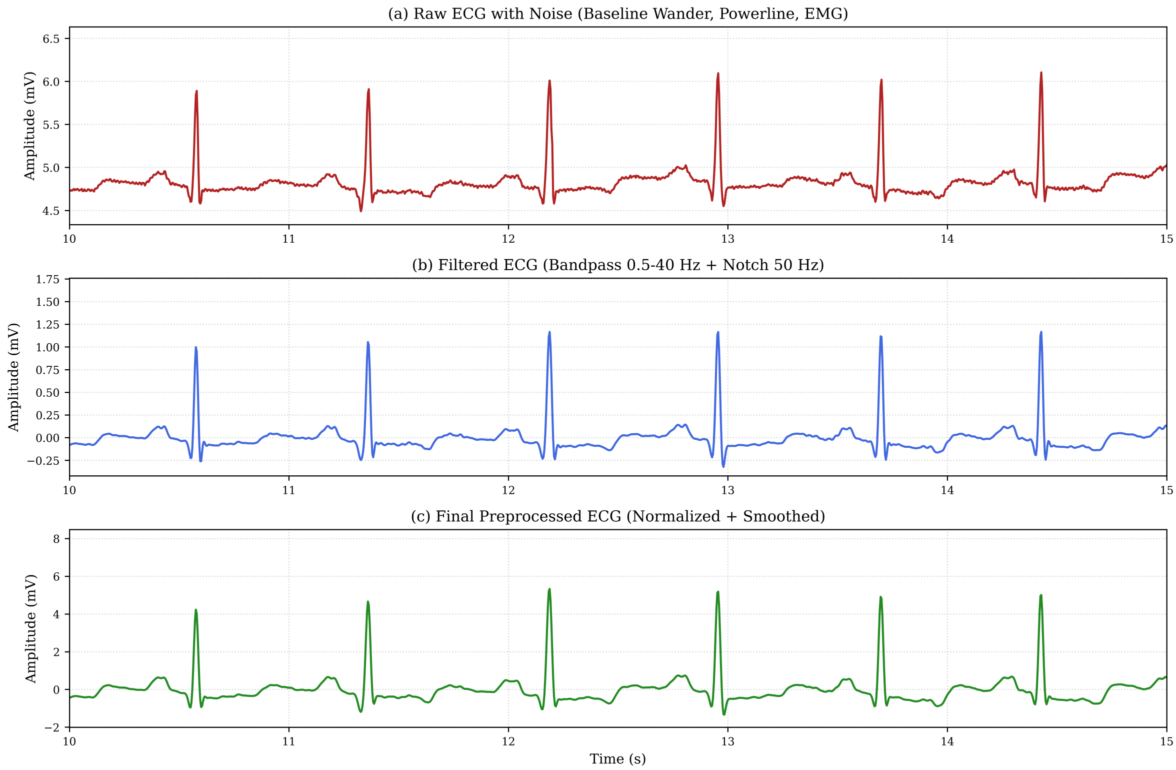


Figure 1.5: Preprocessing stages: Raw, filtered, and normalized ECG signal.

c Detection of P, QRS, and T Waves

The P wave (atrial depolarization), QRS complex (ventricular depolarization), and T wave (ventricular repolarization) form the basis of ECG interpretation [11]. Accurate detection of these waves is essential for understanding how the heart is functioning and spotting potential abnormalities.

Challenges:

- Low P wave amplitude in noisy recordings [32; 37].
- High morphological variability of QRS and T waves across patients [39; 64].
- Overlapping waveforms, especially during tachycardia.
- Signal artifacts like baseline drift or EMG interference [29; 54].
- Differences between datasets that affect how well models generalize [35].

d QRS Detection Algorithms

QRS detection is one of the most studied parts of ECG analysis because of its high amplitude and clinical importance. Classical methods like the Pan-Tompkins algorithm use filtering and thresholding steps to detect QRS complexes efficiently [42]. Other well-known approaches include wavelet transforms [2] and adaptive windowing techniques [34].

More recently, deep learning models have shown superior performance, especially in noisy or highly variable signals:

- **CNNs** extract spatial features that capture the shape and sharpness of the QRS complex [21].
- **LSTMs/RNNs** capture how the signal changes over time, improving temporal understanding.
- **TCNs** and **1D U-Nets** scale well and are commonly used for full-beat segmentation [64].

While classical methods are lightweight and interpretable, deep learning models offer greater flexibility and accuracy in real-world conditions.

e Detection of P and T Waves

P waves help identify issues in the atria, such as atrial fibrillation, while T waves give insight into how the heart recovers after each beat — which is important for spotting repolarization problems [32; 37]. Compared to QRS complexes, P and T waves are often harder to detect because they have lower amplitude and are more affected by noise and signal variation.

Detection Methods:

- **Classical:** Earlier approaches like wavelet transforms, template matching, and baseline correction have been used to detect P and T waves, especially when signals are relatively clean [45].
- **Modern:** Newer deep learning models, including CNNs, improved versions of U-Net with residual connections, and models that use attention mechanisms such as transformers, have shown strong performance even in noisy or irregular signals [16; 25].

1.2.2 Morphological Feature Extraction

Once the P, QRS, and T waves are detected, we can measure specific features from them — these are called morphological features, and they help doctors understand the heart's condition more precisely.

These features include:

- **Durations:** How long each wave lasts (P, QRS, T).
- **Intervals:** The timing between waves (PR, QT, ST, RR), which can reveal conduction problems.
- **Amplitudes:** The height of each wave (P, Q, R, S, T), which reflects signal strength.

- **Ratios:** Relationships like T/R or P/R that can help detect abnormalities.
- **Shape-related metrics:** Slope, area under the curve, and overall waveform morphology.
- **Rhythm indicators:** Features like RR variability, which relate to heartbeat regularity.

Traditional methods calculate these values using signal peaks and thresholds [28]. More advanced techniques apply signal processing tools like wavelet transforms or EMD to improve accuracy [26]. Recently, deep learning models such as CNNs have been used to learn these features directly from the raw signal [52].

1.2.3 ECG Classification

In ECG analysis, classification refers to the process of assigning diagnostic labels to segments or beats of the signal. For example, a beat may be classified as normal or abnormal, helping to detect conditions such as arrhythmias or signs of reduced blood supply to the heart [24; 43].

AI-based classification helps save time for medical staff and makes diagnoses more consistent across different cases. However, challenges remain due to variations between patients, signal noise, and the low frequency of some critical conditions, which makes them more difficult for models to recognize accurately. [3; 35].

Classification Models:

Traditional ML: Older approaches like Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Random Forest models rely on features like wave durations and amplitudes. These models are simple and work well when the input signal is clean and well-preprocessed [24].

Deep Learning: Newer models such as CNNs, RNNs, and Transformers can learn directly from raw ECG signals without needing hand-crafted features. These models are especially powerful when trained on large datasets like MIT-BIH or PTB-XL [3; 43].

Table 1.3: Common Models for ECG Classification

Model Type	Examples	Key Characteristics
Traditional ML	SVM, k-NN, Random Forest	Requires handcrafted features; simple and interpretable.
CNN	ECGNet, ResNet	Learns morphology-based spatial features.
RNN	LSTM, GRU	Models sequential dependencies.
Hybrid	CNN-LSTM, CNN-GRU	Combines spatial and temporal learning.
Transformers	ECGFormer	Captures long-term dependencies and supports interpretability.

Considerations: Deep learning models usually need large labeled datasets and more computing power. On the other hand, traditional ML models are lighter and easier to run, which makes them a better fit for embedded or mobile systems in some cases.

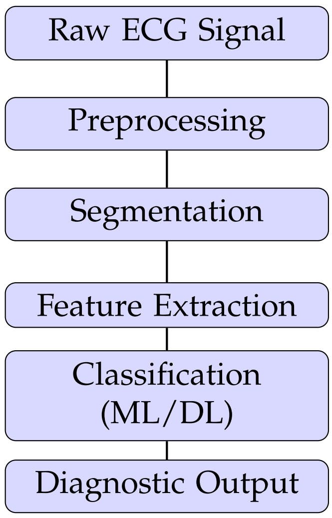


Figure 1.6: Pipeline for Automated ECG Classification

1.2.4 Machine Learning for ECG

a Training and Evaluation

Before an ECG model can be used in practice, it needs to be trained on data and tested to see how well it performs. Popular datasets like MIT-BIH and PTB-XL provide a large number of labeled ECG recordings for this purpose.

To prepare the data, signals are usually cleaned and normalized to reduce noise and ensure consistency. Once trained, models are evaluated using common metrics such as accuracy, sensitivity, and the F1-score, which help assess how well the system can detect different heart conditions [35; 44].

b Deployment and Use

Traditional machine learning models are generally lightweight and fast, which makes them a good choice for situations where computing resources are limited. Deep learning models, while more complex, can still be used in real-time applications if they are properly optimized. With the right balance between performance and efficiency, both types of models can support practical, real-world ECG analysis.

Table 1.4: Comparison Between Traditional Machine Learning and Deep Learning for ECG Analysis

Aspect	Traditional ML	Deep Learning
Feature Extraction	Manual	Automatic from raw ECG
Interpretability	High	Low (black-box)
Computation	Low	High
Performance (Big Data)	Limited	Strong
Deployment	Embedded-friendly	Requires optimization

1.2.5 Model Interpretability and Explainability

In healthcare, it's important not just for a model to give accurate results — but also to understand how and why it made its decision. This is especially true in ECG analysis, where doctors need to verify and trust the system's output before using it in a clinical setting.

Traditional machine learning models, such as decision trees or rule-based classifiers, are easier to interpret because their logic is transparent. On the other hand, deep learning models are often seen as "black boxes" — they can achieve high accuracy, but it's not always clear what features they used to make a decision.

To address this, explainable AI (XAI) techniques have been developed to give insight into deep model behavior. These methods help visualize or highlight which parts of the ECG signal influenced the model's output [30; 40; 60].

Table 1.5: Explainability Methods in ECG Classification

Method	Description	ECG Application
SHAP	Shapley-based feature attribution	Shows which parts of the ECG most influenced the model's decision.
LIME	Local interpretable modeling	Explains single predictions using a simple, approximate model.
Grad-CAM	CNN activation mapping	Creates heatmaps that highlight important regions in the ECG signal.

1.2.6 AI Agents in Healthcare

AI agents are software programs that can make decisions, act on their own, and adapt to new information. In healthcare, these agents are being used to support tasks like monitoring patients, helping with diagnoses, and managing clinical workflows [23].

Key Features

AI agents have several important characteristics:

- **Autonomy:** They can operate without needing constant human input.
- **Reactivity:** They respond to changes in their environment, such as new patient data.
- **Proactivity:** They can take the initiative to reach their goals.
- **Sociality:** They can communicate and collaborate with other agents or systems.

Types of Agents

Different types of agents are used depending on the complexity of the task:

- **Reactive Agents:** These follow simple rules and respond quickly to changes.
- **Cognitive Agents:** These have internal memory and planning abilities.
- **Hybrid Agents:** These combine both reactive and cognitive elements to adapt better in complex environments.

Multi-Agent Systems (MAS)

Multi-agent systems involve several agents working together toward a common goal. In healthcare, this might include agents handling tasks like signal preprocessing, anomaly detection, and report generation. MAS are especially useful when tasks can be distributed or when real-time coordination is needed [51].

One popular agent model is the BDI architecture (Belief–Desire–Intention), which gives agents a goal-driven structure. This makes them more suitable for dynamic and evolving environments like healthcare.

Use Cases and Challenges

AI agents are already being used in healthcare to help monitor patients in real time and support medical decisions. But there are still important challenges to solve — like protecting patient privacy, making sure systems work well together, and proving that these tools are reliable in clinical settings.

1.3 Synthesis and Discussion

This literature review shows how much progress has been made in automatic ECG analysis. Major advances include improved segmentation using models like U-Net [39], more accurate feature extraction based on wavelets and morphology [28], and better classification thanks to deep models like CNNs, LSTMs, and transformers [33]. Even with these improvements, many systems still face obstacles that make clinical use difficult — especially around transparency, smooth integration into workflows, and real-world reliability.

1.3.1 Key Challenges in the Literature

- **Fragmented Processing Pipelines:** Many systems handle ECG analysis as separate steps — like segmentation, feature extraction, and classification — often using different tools for each [49]. This separation can make it hard to optimize the whole system and slows things down in real-time situations.
- **Performance vs. Interpretability:** Deep learning models usually give better accuracy than traditional ones [12; 43], but they often act like "black boxes," making it hard for doctors to understand how decisions are made. Tools like SHAP [40] and LIME help explain predictions [4], but they're not always easy to use in clinical settings.
- **Sensitivity to Noise and Artifacts:** Classic methods like Pan-Tompkins [42] are still used but can fail when the signal is noisy — from patient movement or poor contact [29]. Deep learning models are more resistant to noise, but they need large and varied training datasets to work well across different cases [35].

- **Computational Load:** Some deep models, like LSTM-CNN hybrids [3], perform well but require a lot of computing power, making them hard to run on portable or embedded systems. Simpler models are easier to deploy [44], but they don't always reach the same diagnostic performance.

1.3.2 Emerging Opportunities and Research Directions

Recent research points to several promising ways to move forward:

1. **More Integrated and Dynamic Pipelines:** Some recent studies recommend merging ECG processing steps into a single system where earlier steps inform later ones in real time [21; 64]. This approach better mirrors how doctors interpret ECGs and helps ensure consistency.
2. **Designing Models with Interpretability in Mind:** Rather than just adding explanation tools after the fact, researchers are now trying to build models that are transparent from the start [4; 40]. For example, hybrid approaches that mix expert rules with AI are showing good results for clarity and performance [23].
3. **Lightweight and Modular Architectures:** Multi-agent systems — like those built with the SPADe framework [51] — divide the work between small, specialized components that cooperate. This makes the system more flexible, easier to update, and efficient enough for real-time use.

1.3.3 Summary of Comparative Limitations

The following table compares classical, deep learning, and hybrid methods based on their strengths and weaknesses:

Table 1.6: **Comparison of ECG Analysis Approaches in the Literature**

Aspect	Classical ML	Deep Learning	Emerging Hybrid Approaches
Pipeline Integration	Low (separate steps handled manually)	Medium (all-in-one but hard to interpret)	**High** (well-coordinated modular agents [51])
Interpretability	High (easy to understand rules)	Low (needs extra tools to explain decisions)	**Medium–High** (built-in explainability using agents + SHAP [40])
Noise Robustness	Medium (uses filters to clean signal)	High (learns to ignore noise from data)	**High** (smart filtering combined with deep models [39])
Embedded Feasibility	High (can run on simple devices)	Low (needs a lot of computing power)	**Medium** (efficient agents + lightweight models [44])

1.3.4 Synthesis Summary

In summary, the field of automatic ECG analysis has come a long way. Traditional methods are still useful because they are easy to understand, but they struggle with complexity. Deep learning brings strong performance but raises concerns about trust and technical demands. A new path forward is through hybrid systems — especially modular agent-based designs — that aim to balance performance, clarity, and practical use in clinical settings. This section sets the stage for the next chapter, which presents a multi-agent architecture developed to overcome these challenges.

Conclusion

This chapter explored both the medical and technical foundations of ECG analysis, from understanding how the heart works to the latest methods for segmenting signals, extracting features, and classifying heartbeats. We also looked at how machine learning, explainable AI, and intelligent agent systems are being used to support clinical decision-making. While many of these approaches show promise, challenges like noise sensitivity, real-time processing, and lack of interpretability still remain. These issues open the door for new solutions, which we begin to address in the next chapter through a modular, agent-based ECG interpretation system.

Chapter

2 Contributions

Introduction

This chapter presents the main contribution of this work: design and development of a real -time ECG interpretation system based on intelligent agents and machine learning models. In the previous chapter, the boundaries identified like lack of modularity, limited real-time capabilities, and ambiguity of traditional black-box models. The purpose of our contribution is to provide a **transparent**, scalable and intelligent approach to automated ECG analysis.

The proposed system is structured around a **multi-agent architecture** [51] that mimics the workflow of human ECG interpretation. Each agent is responsible for a specific function in the pipeline, from signal acquisition and mask detection to features extraction and beat classification then the full signal. These agents are coordinated through a central controller and communicate via **SPADe** framework [41] using the **XMPP** protocol[48].

To increase diagnostic accuracy, several machine learning and deep learning models are integrated into the system, including **CNN** for peak detection, a **UNET-based** segmentation model [39] and a **Random Forest** classifier for beat-level and signal-level classification [44]. The entire architecture is implemented following **Clean Architecture** principles which ensures maintenance, testing and concerns isolation.

This chapter is structured as follows:

- Section 2.1 presents theoretical proposal including system architecture, agent roles and integration of AI models.
- Section 2.2 gives details of the implementation process, used libraries and experiments to validate the system.

2.1 Theoretical Proposal

2.1.1 System Overview

The proposed system is an intelligent and modular platform designed for real-time ECG interpretation. Its primary objective is to automate the full processing pipeline -from acquisition to classification- while maintaining accuracy, scalability and interpretability. This approach addresses the boundaries discussed in the previous chapter, including lack of modularity in traditional systems, insufficient real-time support and limited transparency in the black-box models [40; 51].

The system architecture is structured around three main layers: a user interface (UI), a backend server and a distributed multi-agent system. The UI built using **React**, allows users to send ECG files or broadcast live data through the Websocket. It also serves as a display interface for the signal analyses and final classification results. Communication between the UI website and the backend server is handled via by **RESTful** APIs to upload files, and via **webSocket** endpoints for live detection and analysis [23].

The backend is developed using **Django** and hosts a set of smart agents implemented through the **SPADe** framework [51]. These agents communicate through the **XMPP** protocol using structured **JSON** messages, allowing the data processing to be asynchronous and decentralized data. In the center of the system, there is a controlling agent, which acts as an orchestrator following a **BPMN-style** flow. It dynamically manages the execution of agents in a definite sequence, ensuring the correct spread of data and coordinating responses between agents and the user interface.

Each agent in the system performs a well-defined task in the ECG analysis pipeline. The acquisition agent begins the process of capturing the entry of ECG files. The **segmentation agent** applies a deep learning model to detect P, QRS and T waves [39]. **Post-processing agent** to fix this detected mask. **Features extraction agent** refine the signal and extract clinically relevant measurements. A set of **classification agents** sequentially analyzes the features: The beat classifier labels the individual heartbeat, the signal classifier determines whether the ECG is normal or abnormal, and a specialized agent categorizes abnormal signals. Finally, the **storage agent** records all results in a local **SQLite** database for file and recovery.

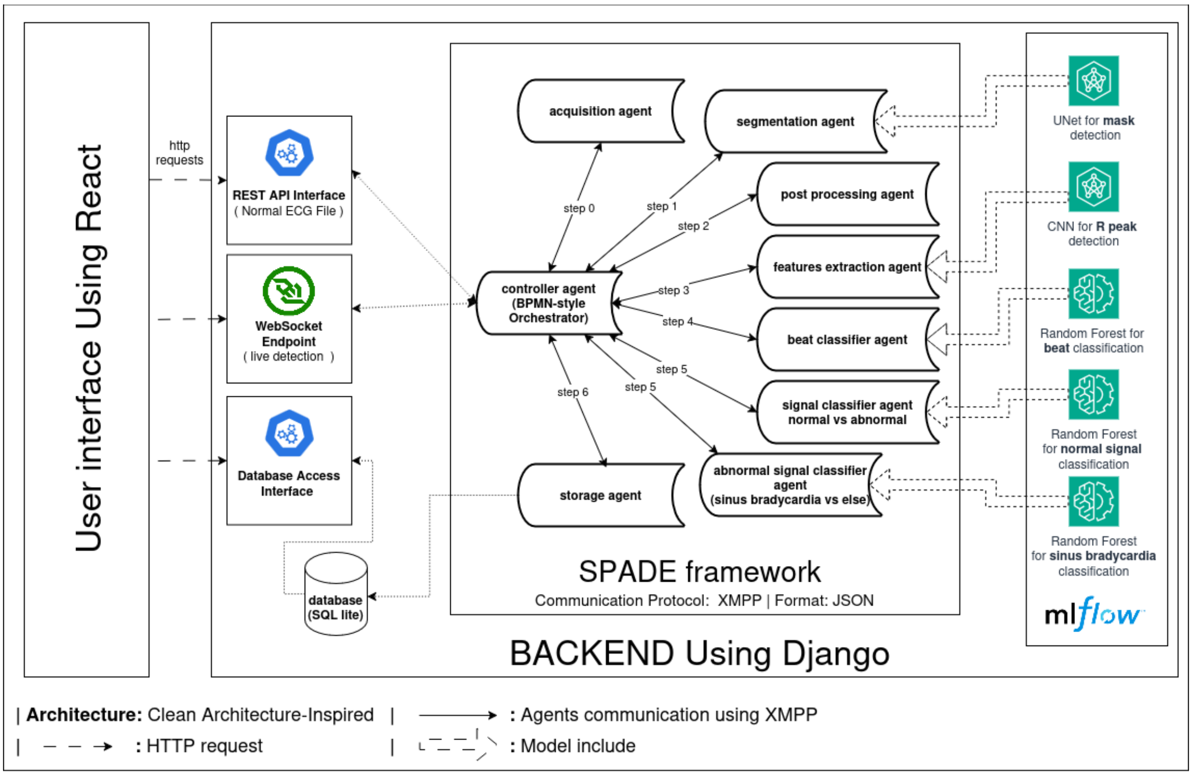


Figure 2.1: System architecture for real-time ECG interpretation based on multi-agent coordination.

A key characteristic of this architecture is its modularity. Each component works independently and can be updated, replaced or scale without disrupting the entire system. This design follows **Clean Architecture** principles, where presentation, domains and infrastructure layers are clearly separated [51]. Such a structure supports long-term stability, facilitates debugging and testing, and and enhances real-time response – a critical requirement in embedded or clinical monitoring environments.

Overall, this system integrates machine learning and deep learning models within a multi-agent coordination structure to provide robust, interpretable and scalable ECG interpretation.

2.1.2 Agent-Based Architecture

Our system relies on a multi-agent design to process ECG signals step-by-step in a modular and coordinated way. Each agent is responsible for a specific task in the pipeline, and they communicate using the XMPP protocol, exchanging data in JSON format. This setup makes the system flexible, easier to maintain, and well-suited for real-time operation [23; 51].

There are nine agents in total. Each one plays a focused role, and they all work together under the supervision of a controller agent. The controller ensures that everything runs in the correct order and that results are sent back to the frontend when ready. Below is an overview of each agent and what it does:

- **Acquisition Agent:** Receives ECG input from the user. This can be a static file (uploaded via REST) or live data (via WebSocket). Once received, the raw signal is passed to the segmentation stage [29; 54].
- **Segmentation Agent:** Using a 1D U-Net model [39; 64] to identify the locations of P waves, QRS complexes, and T waves. It outputs a mask that labels each waveform component.
- **Post-Processing Agent:** Cleans the segmented output by removing noise and normalizing the signal. This step prepares the data for accurate and consistent feature extraction.
- **Feature Extraction Agent:** performs beat-by-beat analysis to extract useful measurements from the ECG, such as wave durations, amplitudes, and intervals (PR, QRS, QT, RR) ...etc [28; 52].
- **Beat Classifier Agent:** Classifies individual heartbeats using a Random Forest model [43]. It labels each beat based on its extracted features.
- **Signal Classifier Agent:** Looks at the ECG as a whole and decides whether it is normal or abnormal. If it's normal, the process ends here. If not, another agent takes over.
- **Abnormal Signal Classifier Agent:** Further analyzes abnormal signals to identify specific types of arrhythmias, like sinus bradycardia or atrial fibrillation [12].
- **Storage Agent:** Saves the results (including normalized signal, mask, features, and diseases) into a local SQLite database for future retrieval and analysis.
- **Controller Agent:** Coordinates the entire system. It tells each agent when to run, handles any errors and ensures that the final diagnosis reaches the front end [51].

This agent-based setup makes the system modular and easy to scale. If one agent needs to be improved or replaced, it can be done without affecting the others. It also allows for better error handling and debugging, since each step is isolated. This design fits well with Clean Architecture principles and supports real-time execution in clinical or embedded environments.

Figure 2.2 shows the full agent flow in our system. Each major block—such as classification or segmentation—can also be viewed in more detail through specific diagrams shown later in this chapter.

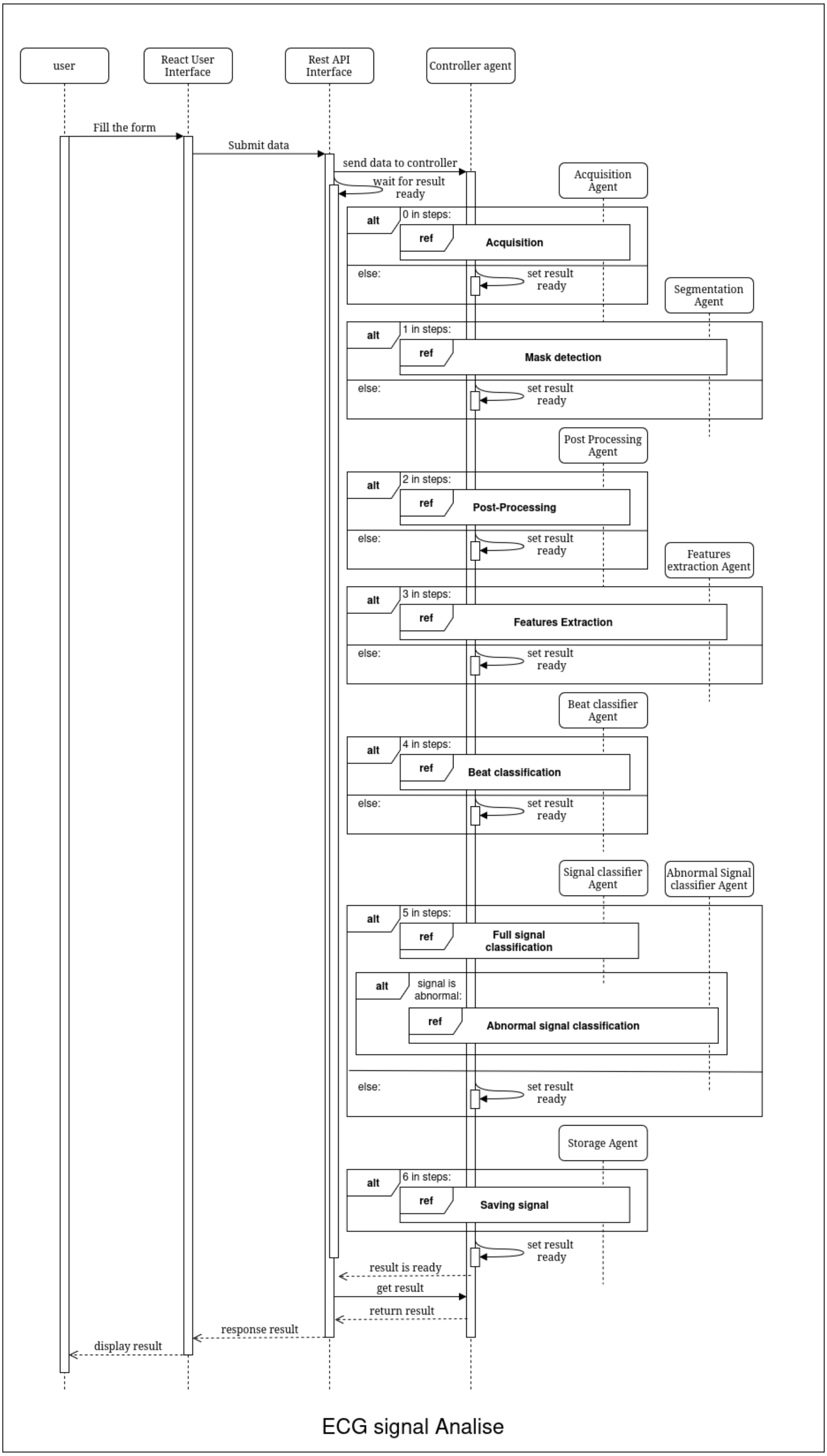


Figure 2.2: Full agent sequence for ECG interpretation. Each block represents a specialized agent.

2.1.3 SPADe Framework and Agent Communication

To implement our multi-agent system, we used the SPADe (Smart Python Agent Development Environment) framework. SPADe is built around the **XMPP** (Extensible Messaging and Presence Protocol), a lightweight and extensible protocol originally designed for instant messaging, now widely used in agent-based systems [48]. This choice supports asynchronous communication, modularity, and real-time message exchange between agents [41].

Each agent in our architecture is implemented as a SPADe agent with a unique XMPP identifier (JID) and a defined behavior. Agents run concurrently and independently, exchanging information using structured **JSON** messages over the XMPP protocol. This architecture ensures that agents can be distributed across machines, restarted individually, or extended without modifying the rest of the system [47].

Why SPADe and XMPP?

We used SPADe because it makes it easier to build and manage agents using the XMPP protocol. It supports sending messages between agents, reacting to events, and works well with Python and machine learning tools. This helped us create a system that is simple, flexible, and works in real time [41].

Using XMPP allows the system components to work independently from each other. Messages can be delivered reliably and securely over local networks or the internet [48]. JSON was chosen as the data format because it is lightweight, human-readable, and easy to parse across platforms, including web clients (React frontend), backend services (Django), and agents (Python) [50].

Setting Up the Agents

During testing, all agents were run on a local Openfire XMPP server. Each one had its own login and was set up to handle a specific task in the ECG pipeline, like segmentation, classification, or feature extraction. A special agent, called the Controller, was in charge of coordinating the work. It made sure tasks ran in the right order, messages were sent to the correct agents, and the system could recover if something went wrong. This setup made the system easier to manage, test, and update.

Benefits of This Framework

This design offers several practical benefits:

- **Scalability:** New agents can be added without modifying the rest of the system.

- **Modularity:** Each agent has a single responsibility and can be developed or tested independently.
- **Resilience:** If an agent fails, the controller can isolate the error and continue the workflow.
- **Real-Time Compatibility:** Supports both batch processing and live streaming use cases.

2.1.4 Model Choices

Our approach is based on combining deep learning with classical machine learning methods to segment the ECG signal, extract features and thereafter recognise the class of the incoming signal. These design decisions were motivated by a tradeoff between performance and interpretability given real-time efficiency of the method in clinical settings.

Segmentation Models: From CNN-LSTM and TCN to U-Net

During development, we had tried several deep learning architectures to segment ECGs, such as the hybrid CNN-LSTM networks [3] and TCNs [6]. This approach performed well, but failed to properly identify fine waveform boundaries, especially in noisy signals and at boundary cases where waves overlap one another.

We finally chose a 1D U-Net model [39; 64] for the segmentation application. The U-Net performed better than other models mainly because of its encoder-decoder design and skip connections that enabled the model to integrate low-level detail with high-level context. This is important for adequate P, QRS and T wave deliniation in a variety of datasets.

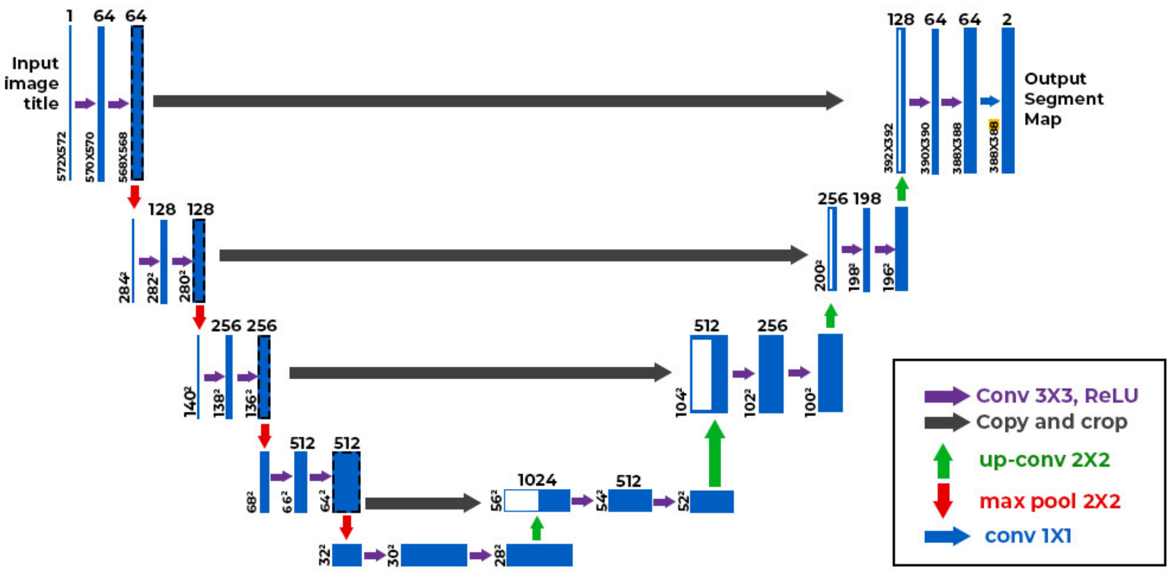


Figure 2.3: U-Net Architecture.

For the success of the model, we have particularly tackled the problem of class imbalance in ECG which is severe, because the background points are orders of magnitude larger than the waveforms in the ECG. We utilized a weighted **Focal Loss** based model to train the model, that prioritizes the focus on "hard" wave boundaries and the under-represented wave classes. [1; 63]. The performance of this segmentation approach was evaluated by **the Intersection over Union (IoU)** measure more suitable to the edge-based accuracy[39].

R-Peak Detection: CNN-Based Model

Besides full-wave segmentation, we further comprised a light-weight CNN model for accurate R-peak detection. This aid is important for the alignment of the beats, the computation of RR intervals and the validation of QRS boundaries [42]. The CNN strategy was selected in our work as it can precisely measure local maxima with low timing cost.

Classification Models: Random Forest Cascade

For classification, We used a cascade of Random Forest models [24; 43]. These classifiers were developed from analyzed morphological parameters, aimed at improving the diagnosis:

- **Beat Classifier:** Classifies each beat using metrics such as QRS-duration, PR interval, amplitude of waves.
- **Signal Classifier:** Evaluates the full signal to determine whether it is normal or abnormal.
- **Abnormal Signal Classifier:** Further categorize abnormal signalsas particular arrhythmias (e.g., sinus bradycardia).

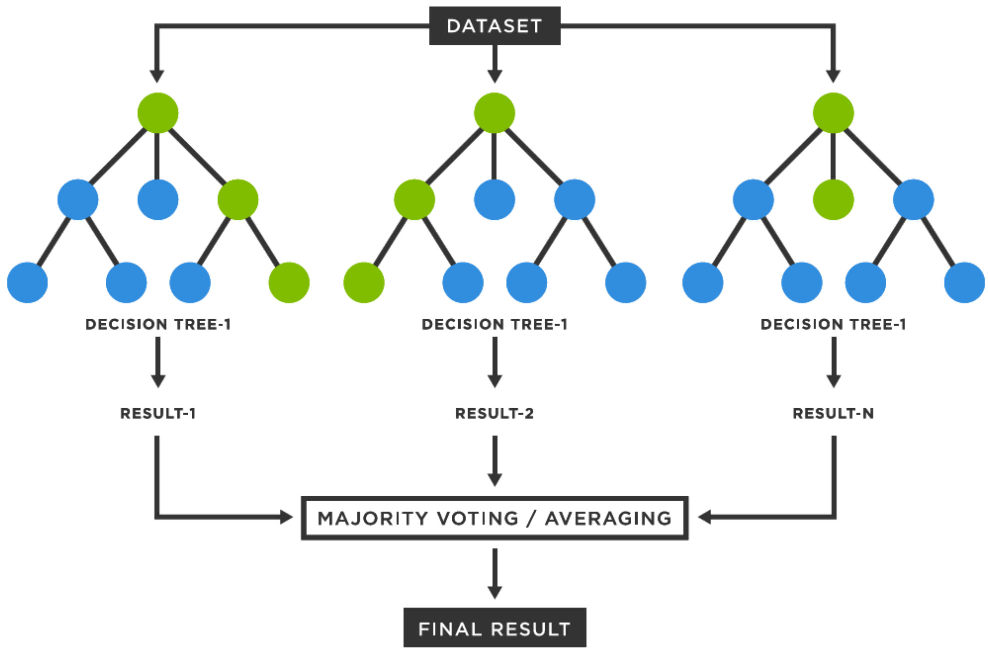


Figure 2.4: Random Forest.

Why This Hybrid Strategy?

U-Net was chosen for segmentation as this demonstrated the best generalization over signal type as well as the best preservation of high-frequency wave details. CNNs were conducive to the fast and accurate detection of R-peak. Random Forest classifiers were chosen for their interpretability, speed, and accuracy on small feature spaces. [12; 35].

Summary of Model Stack

- **Segmentation:** U-Net (final), tested against CNN-LSTM, TCN
- **R-peak Detection:** CNN
- **Classification:** Three-stage Random Forest pipeline

Such a mixed modeling method guarantees that our system can keep its strength in robustness, interpretability, denoising efficacy for both the batch and real-time ECG analysis.

Summary

We have described in this section the theoretical basis of our proposed system for real-time interpretation of ECG. We started with a global view of the pipeline followed by the agent-based architecture ensuring both modularity and coordination. We motivated our choices of models in terms of a trade-off between performance and interpretability, and we motivated our use of the SPADe framework and the XMPP protocol between the agents for communication. Collectively, these design choices strike a balance among deep learning accuracy, machine learning interpretability and system-level flexibility — the foundational pillars for constructing a scalable and clinically trusted platform for ECG analysis.

2.2 Implementation and Experiments

2.2.1 Dataset

Our system was built using several open-access ECG datasets selected for their variety, size, and medical relevance. These datasets were used for waveform segmentation, beat classification, and full signal classification. All datasets were processed using the same cleaning and formatting steps to make sure the models could work together smoothly.

The datasets were organized by the role they played in the system:

- **Waveform Segmentation: LUDB and QTDB**
  The Lobachevsky University Electrocardiography Database (LUDB) contains 200 12-lead ECG recordings with expert annotations for P, QRS, and T waves [27]. The QT Database (QTDB) includes 105 two-lead recordings with detailed waveform labels [31].
- **Beat Classification: MIT-BIH Arrhythmia Database (MITDB)**
  The MIT-BIH Arrhythmia Database has 48 long two-lead ECGs with over 110,000 labeled beats, including many arrhythmia types, and is a standard benchmark for arrhythmia classifiers [18; 38].
- **Signal Classification: PhysioNet Challenges and others**
  Data from multiple PhysioNet Challenges were used, including the 2021 Challenge (over 88,000 ECGs) [46] and the 2017 Challenge (8,528 short ECGs) [10]. We also used the Normal Sinus Rhythm Database (NSRDB) [17] and the ECG arrhythmiaDatabase (ecg-arrhythmia) [65].
- **Testing: European ST-T Database (EDB)**
  The European ST-T Database contains 90 excerpts with annotated ST and T wave changes, making it suitable for testing ischemia detection [55].

These datasets cover a wide range of ECG patterns and patient types, helping improve the generalization of the system.

2.2.2 Signal Preprocessing Pipeline

All ECG signals were preprocessed using the same pipeline implemented in a Kaggle notebook[1]. The main steps were:

- **Bandpass Filtering:** A zero-phase Butterworth bandpass filter (0.5–15 Hz) was applied to remove baseline drift and muscle noise [29].
- **Smoothing:** A moving average filter was used to reduce residual high-frequency noise [54].
- **Resampling:** To ensure consistency, all signals were resampled to a uniform sampling frequency of 250 Hz [54].
- **Normalization:** Finally, z-score normalization was applied to center the signal's baseline and scale its amplitude, providing a standardized input for the machine learning models [54].

This standard preprocessing ensured compatibility across all parts of the pipeline.

2.2.3 Segmentation

For the problem of waveforms segmentation, we compared several deep learning models using the LUDB and QTDB datasets. All input signals were preprocessed into fixed segments with the binary masks for the P, QRS, and T waves. padding was used as necessary, while we also applied simple data augmentation methods, including the addition of noise and time warp, to increase the model's generalizability. [64].

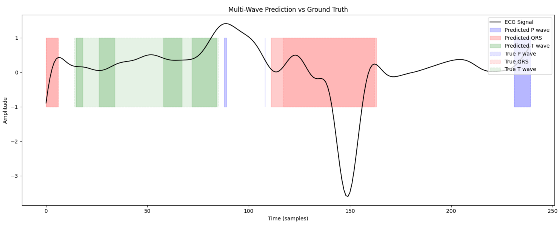
We tried three main architectures: a Temporal Convolutional Network (TCN)[2] [6], a hybrid CNN-LSTM[3] network commonly used for sequential data [3], and a 1D U-Net[4] [39]. In order to overcome the class imbalance present in the ECG data, we trained our final model using a weighted **Focal Loss** which was developed to perform well on an imbalanced set of categories [63]. Their results were measured according to the **Intersection over Union (IoU)** accuracy metric as commonly used for segmentation.[39].

The 1D U-Net architecture finally performed best, particularly with the finer P and T wave delineation, which validates its nomination for our final pipeline. Comparative results are shown in Table 2.1 and Figure 2.5.

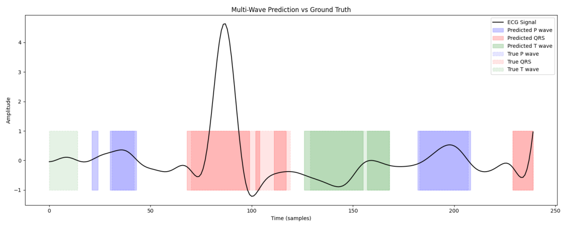
[1] https://www.kaggle.com/code/abdessamiguebli/signal-pre-processing
[2] https://www.kaggle.com/code/abdessamiguebli/tcn-for-ecg-mask-detection
[3] https://www.kaggle.com/code/abdessamiguebli/cnn-lstm-for-ecg-mask-detection
[4] https://www.kaggle.com/code/abdessamiguebli/test-unet-for-ecg-mask-detection

Table 2.1: Comparison of Segmentation Model Performance using Mean IoU

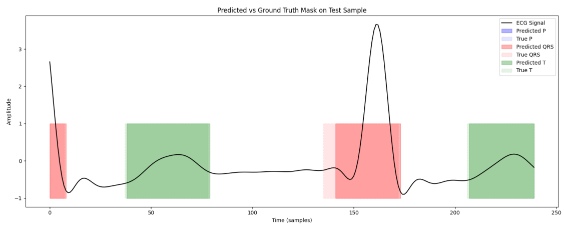
Model	Train Acc	Train Loss	Val Acc	Val Loss	Test Acc	Test Loss
TCN	0.9560	0.1081	0.9569	0.1075	0.9563	0.1091
CNN-LSTM	0.9433	0.1466	0.9433	0.1461	0.9439	0.1444
1D U-Net (focal loss)	**0.9385**	**0.0106**	**0.9396**	**0.0106**	**0.9413**	**0.0107**



(a) TCN Model



(b) CNN-LSTM Model



(c) 1D U-Net Model

Figure 2.5: Visual comparison of segmentation performance for the TCN, CNN-LSTM, and 1D U-Net models. The U-Net model shows a closer alignment between predicted masks and the ground truth.

The primary evaluation metric for this task was the Intersection over Union (IoU), which measures boundary accuracy. As shown in Table 2.2, the U-Net model demonstrated excellent performance across all wave classes, confirming its suitability for our pipeline.

Table 2.2: Class-wise IoU Performance for the 1D U-Net Model on the Test Set

Waveform/Class	Intersection over Union (IoU)
Background	0.8887
P-wave	0.8367
QRS-complex	0.9307
T-wave	0.8838

2.2.4 Post-processing

After the U-Net model is first used to perform the segmentation, a specially designed Post-processing Agent runs a pipeline of instant refinement operations to correct possible mistakes and to validate that the output mask is anatomically consistent and physiologically plausible. This stage is important to increase the robustness to feature extraction and classification steps. The agent consists of a cascade of rule-based functions:

Initially, the function **`remove_uncompleted_first_last_wave`** is applied to cut the edge of signal, the mask need start and end with background class to avoid dealing with unfinished wave. Next, **`merge_close_waves`** function fixes these fragmentation errors by joining waves of the same type and with small, erroneous gaps between them.

Second, the physiological logic is applied. The function **`remove_irrelevant_waves`** adjusts the sequence to make sure we start recording with a p-wave (if any) and terminate with a t-wave and it removes leading and trailing artifacts. The **`check_repeated_waves`** algorithm then resolves ambiguities, such as two consecutive P-waves, by removing the segment with weaker morphological characteristics.

A refinement is then applied to adjust the boundary in a fine-grained manner. The functions **`fix_P`** and **`fix_QRS`** treat the local peaks and slopes of the signal to properly expand or reduce the start and end of the wave detections. This guarantees that the complete morphology of each wave is correctly represented prior to feature extraction.

2.2.5 Features Extraction

a R-peak Detection

After full-wave segmentation, a lightweight 1D Convolutional Neural Network (CNN) was developed for the precise location of R peak in each beat[5]. The use of CNNs for feature detection in ECG signals is a well-established and effective approach [64]. Accurate R-peak detection is fundamental for calculating critical rhythm-based features, such as RR intervals and heart rate variability.

The model uses a simple encoder-decoder structure. The encoder path consists of two `Conv1D` blocks with Batch Normalization and ReLU activation, which extract features from the 250-sample input window. The decoder path then upsamples these features, and a final convolution produces a time-series output that indicates the probability of an R peak at each sample. The model architecture is summarized below:

[5] `https://www.kaggle.com/code/abdessamiguebli/r-detection`

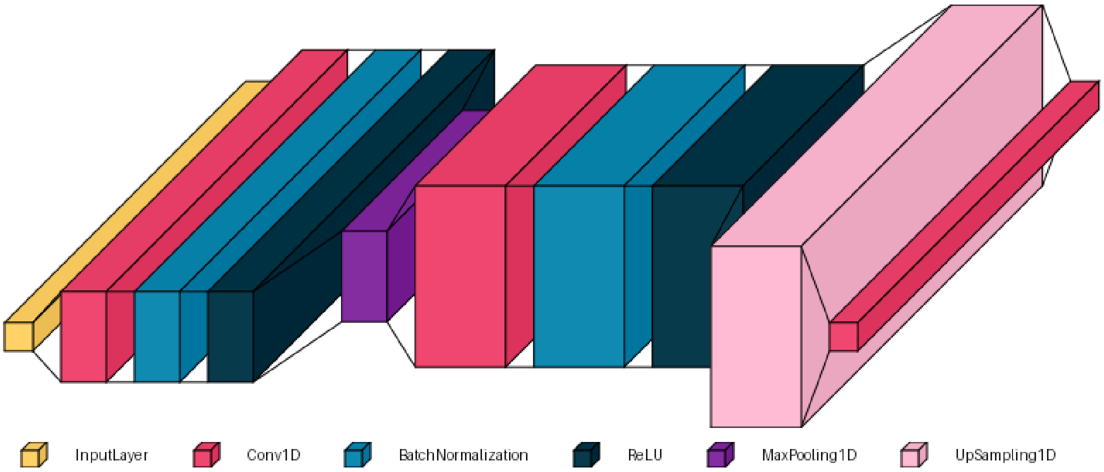


Figure 2.6: Architecture of the 1D CNN model used for R-peak detection.

The model's performance was outstanding, achieving a validation accuracy of **99.60%** with a validation loss of just **0.0083**. This high level of accuracy ensures that the R-peaks are reliably detected, providing a solid foundation for all subsequent feature extraction tasks.

b Features Extraction

After segmentation, the Feature Extraction Agent computed a comprehensive set of key ECG features for each beat, drawing from established morphological and rhythm-based metrics commonly used in automated ECG analysis [28; 52]. The extracted features included the following:

- **Morphological Features:** Features describing the shape of individual waveforms.
  - *Durations:* P-wave, QRS complex, and T-wave durations.
  - *Intervals:* PR, QT, and ST intervals.
  - *Amplitudes:* P, Q, R, S, and T-wave peak amplitudes.
  - *Ratios:* T/R and P/R amplitude ratios.
  - *Slopes:* QR and RS slopes within the QRS complex.
  - *Axis:* QRS axis estimation.
  - *Shape:* P-wave symmetry and T-wave inversion.
- **Rhythm and Variability Features:** Features describing the timing and pattern of heartbeats.
  - *Heart Rate:* RR intervals and the corresponding beats per minute (BPM).
  - *Variability:* Local RR variability and the Root Mean Square of Successive Differences (RMSSD).
  - *Arrhythmia Patterns:* Detection of premature beats, bigeminy, and trigeminy.

This rich feature set was then used as the input for the downstream beat-level and full-signal classification models [44].

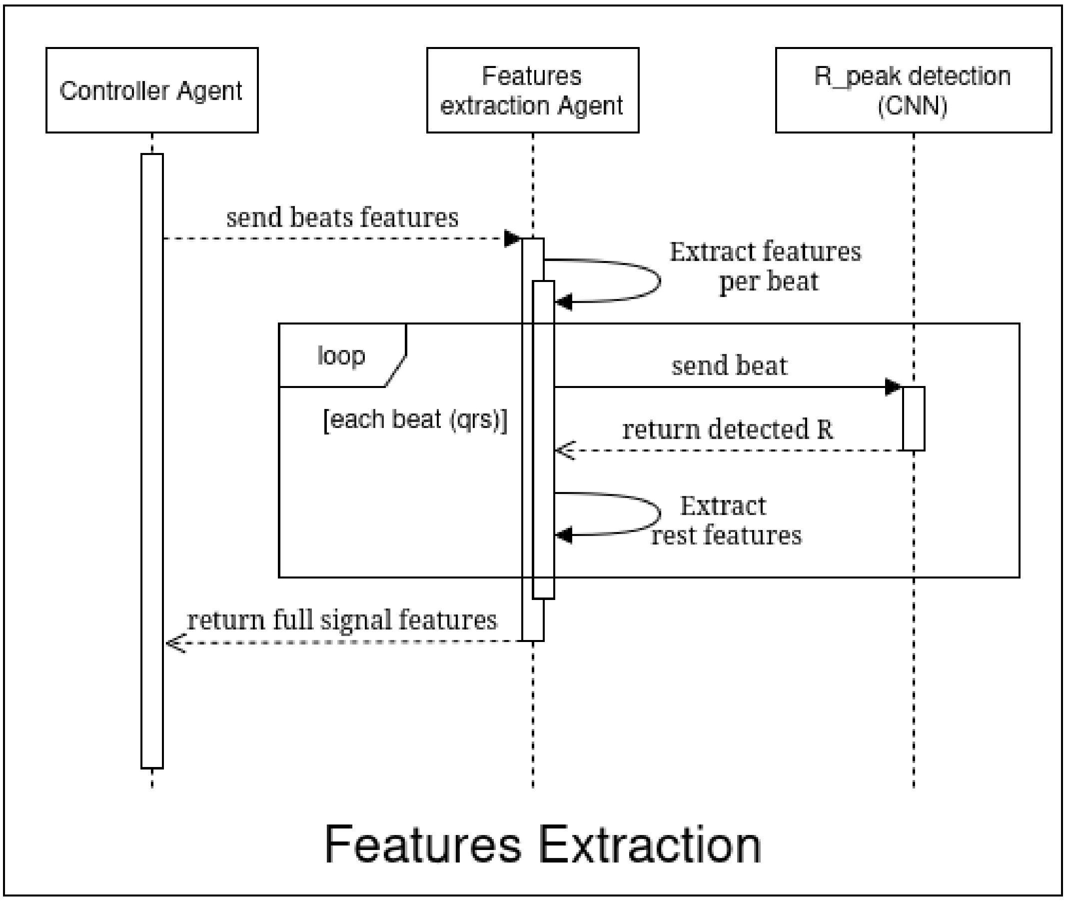


Figure 2.7: Sequence diagram illustrating feature extraction.

2.2.6 Beat Classification

For the beat-level classification task, we used the well-annotated MIT-BIH Arrhythmia Database [18; 38]. From the morphological and rhythm-based features extracted in the previous stage, we constructed a feature set for each individual heartbeat.

A Random Forest classifier, a machine learning model was well adapted for this type of work [24], was trained on this feature set to distinguish between different beat types[6]. The model demonstrated strong performance across all classes, as detailed in the classification report (Table 2.3) and the confusion matrix (Figure 2.8).

[6] `https://www.kaggle.com/code/abdessamiguebli/ecg-beat-classification`

Table 2.3: Beat Classification Performance Report

Beat Type	Precision	Recall	F1-score	Support
Normal (N)	0.98	1.00	0.99	35453
LBBB (L)	1.00	0.94	0.97	1601
RBBB (R)	0.99	0.97	0.98	1584
Paced (/)	0.99	0.98	0.99	1768
Ventricular (V)	0.94	0.93	0.93	1673
Other (Rest)	0.94	0.75	0.83	1662

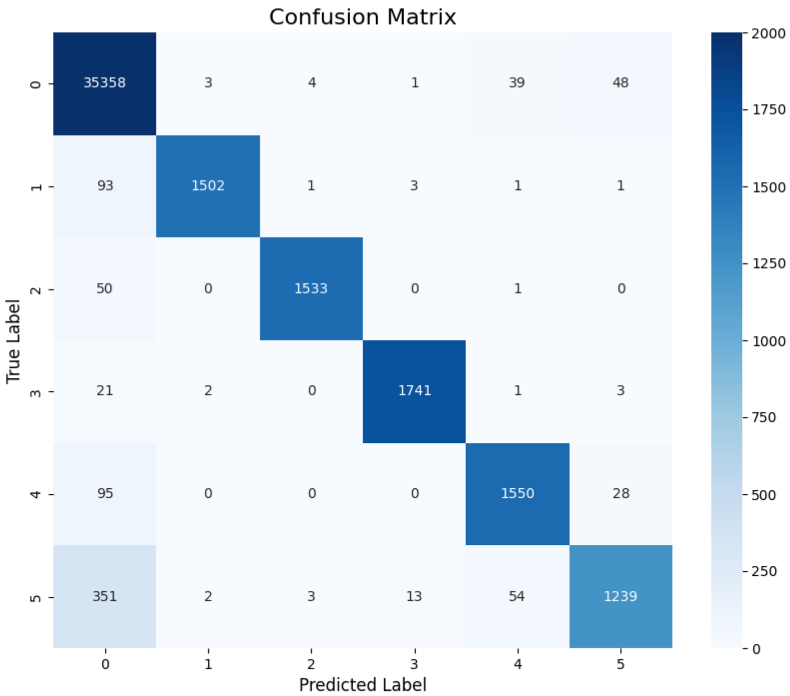


Figure 2.8: Confusion matrix for the Random Forest beat classifier.

The model achieves near-perfect precision and recall for Normal beats and demonstrates high F1-scores for the various arrhythmia classes, confirming its reliability for use within our diagnostic agent pipeline.

2.2.7 Full Signal Classification

The final stage of the pipeline involves classifying the entire ECG signal to provide an overall diagnostic impression. This is accomplished in a two-step process: collecting beat-level features into a signal-level summary and then feeding this summary into a final classification model.

a Full Signal Feature Extraction

To create a comprehensive representation of the entire ECG recording, the beat-by-beat features extracted earlier are aggregated into a single feature vector. For each morphological and rhythm-based metric (e.g., QRS duration, P-wave duration, heart rate, RMSSD), we calculate a set of statistical properties: the **mean**, **standard deviation**, **minimum**, and **maximum** values across all detected beats in the signal. This process of transforming time-series data into a fixed-size statistical summary is a common technique for preparing ECG data for classification [52]. This feature vector captures both the central tendencies and the variability of the cardiac cycle, which is essential for identifying persistent abnormalities.

b Normal vs. Abnormal Classification Model

Using the collected signal features, a final Random Forest model was trained to perform binary classification[7], labeling each ECG recording as either Normal (Class 0) or Abnormal (Class 1). This model was trained and validated on a large composite dataset of **126,857** signals, drawn from sources including the PhysioNet 2017 Challenge [10], the PhysioNet 2021 Challenge [46], the Normal Sinus Rhythm Database (NSRDB) [17], and the ECG Arrhythmia Database [65].

The model achieved a robust overall accuracy of **90.77%**. The detailed performance, including a high recall for the abnormal class, is presented in the classification report (Table 2.4), confirming its effectiveness in correctly identifying pathological signals.

Table 2.4: Normal vs. Abnormal Signal Classification Report

Class	Precision	Recall	F1-score	Support
Normal (0)	0.92	0.85	0.89	10564
Abnormal (1)	0.90	0.95	0.92	14808
Weighted Avg	**0.91**	**0.91**	**0.91**	**25372**

Table 2.5: Abnormal classification metrics

Metric	Formula	Value	Goal
Recall (Sensitivity)	$\frac{TP}{TP+FN}$	0.9460	≥ 0.95
Precision	$\frac{TP}{TP+FP}$	0.9013	> 0.90
F1-Score	$2 \times \frac{P \cdot R}{P+R}$	0.9229	> 0.90
Specificity	$\frac{TN}{TN+FP}$	0.9187	> 0.85

[7] https://www.kaggle.com/code/abdessamiguebli/full-signal-classification-model

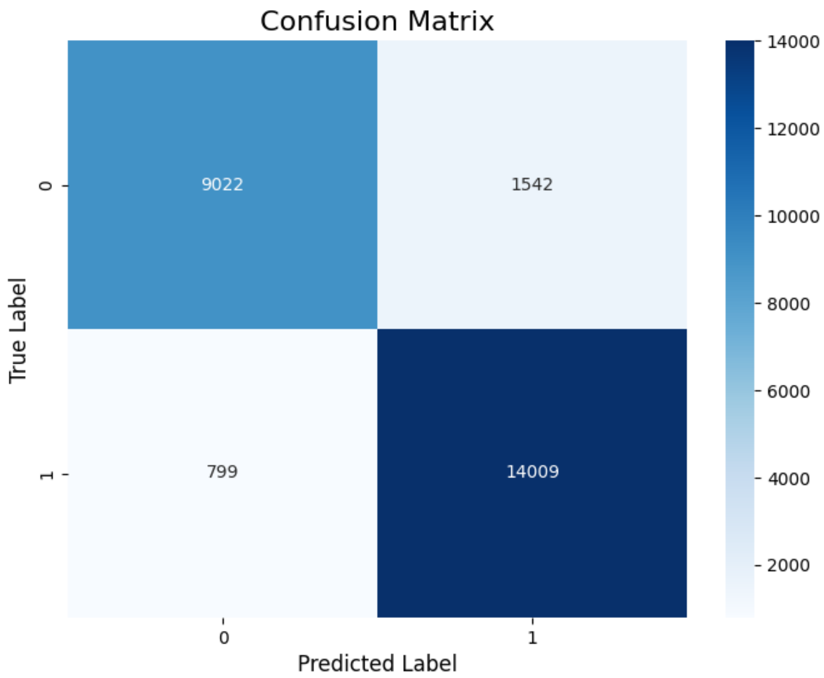


Figure 2.9: Confusion matrix for the Random Forest Full Signal classifier.

c Abnormal Signal Classification

Once a signal is identified as abnormal, a further layer of analysis is triggered to screen for specific pathologies. For this purpose, we trained a series of independent, binary Random Forest models[8], each dedicated to detecting a single disease. This modular approach, often referred to as binary relevance, is a common strategy for handling multi-label classification problems and allows for targeted training and evaluation for each condition [43]. The performance of the initial four disease-specific models is summarized in Table 2.6.

Table 2.6: Performance Summary of Disease-Specific Classification Models

Condition	Training Set Size	Accuracy	F1-score (Disease)
Sinus Bradycardia	59,000	0.9719	0.98
Sinus Tachycardia	24,732	0.9616	0.96
Supraventricular Tachycardia	2,106	0.9597	0.96
Sinus Arrhythmia	8,188	0.9170	0.92

[8] https://www.kaggle.com/code/abdessamiguebli/other-diases-models

2.2.8 System Architecture and Integration

To operationalize the classification pipeline, we created a full backend system, which includes trained machine learning models as part of a reliable, scalable and real-time processing environment. The system is based on a modular architecture and uses a multi-agent system for its processing, a web framework as UI, and specialized tools for model and data handling.

a Backend Framework and Database

At the core of all this, there is the **Django REST Framework** that we use to build secure and flexible Web **APIs**. All data was dumped (features, predictions and signal metadata) into a **SQLlite** database. This relational database allows for data normalization, efficient access via its own API endpoint and enables us to further store and retrieve past analyses.

b Multi-Agent System for ECG Processing

The system intelligence is modulated by a **Multi-Agent System (MAS)** built with the **SPADe** framework [41; 47]. Every step of the ECG analysis pipeline, ranging from segmentation to classification, is encapsulated in its own autonomous agent. These agents are implemented as asynchronous communicating systems based on the **Extensible Messaging and Presence Protocol (XMPP)** [48], exchanging data in the form of structured **JSON** messages [50]. This architecture provides a high degree of modularity in each part of the system and freedom to evolve and scale each appropriately while remaining robust against failures in individual parts of the pipeline. server served as the centralized server for the entire MAS.

c Model Serving and Management

All the trained machine learning models are organized and served by **MLflow**, an open-source platform for the machine learning lifecycle. MLflow's tracking and registry features enable versioning, deploying, and monitoring models to create a complete prediction workflow that is repeatable and supportable.

d API and Real-Time Communication

The interaction system reveals two main modalities:

- **REST API for Batch Analysis:** A standard API endpoint allows users to upload an ECG file for a complete, asynchronous analysis. The system processes the signal through the multi-agent pipeline and returns the final diagnosis.

- **WebSocket for Live Detection:** For real-time monitoring, a WebSocket connection is established. This channel allows for the continuous streaming of ECG data, which is processed in segments (e.g., every 15 seconds). The system provides live feedback and classification results through this persistent connection, enabling immediate clinical decision support.

2.2.9 User Interface

The frontend was developed using the React.js framework to provide a clean and interactive interface for analyzing ECG signals. The core functionality is accessible via the "ECG Analysis" page, where users can upload signal files, run the backend classification pipeline, and view results in real time, as shown in Figure 2.10.

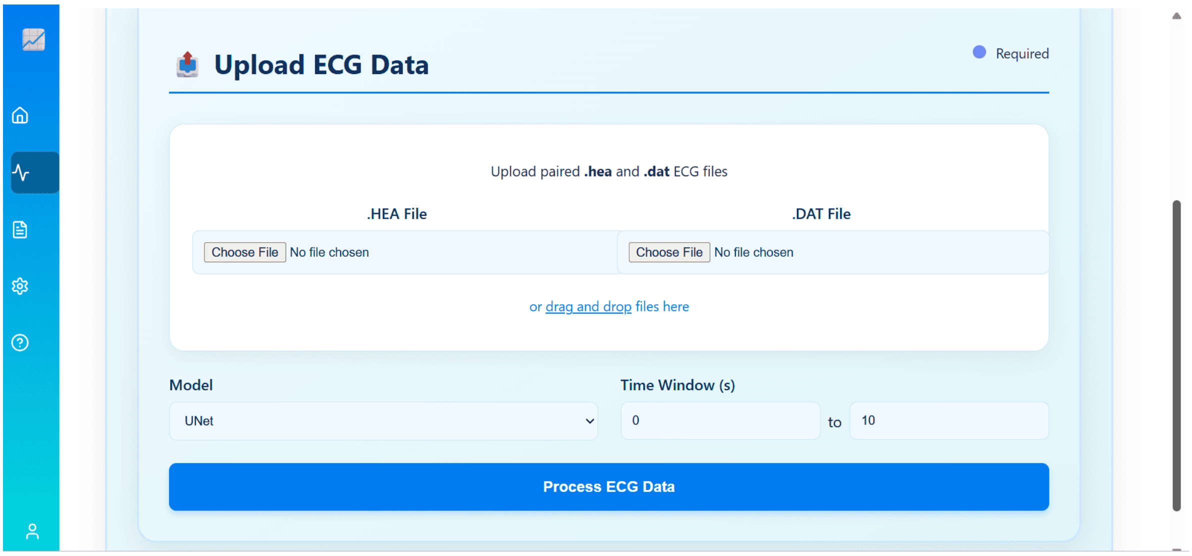


Figure 2.10: Upload interface for ECG data files and model selection.

a ECG Analysis Page

The ECG Analysis page allows users to upload ECG records in `.dat` and `.hea` format. After specifying a signal window and selecting a segmentation model (e.g., UNet), the system sends a request to the backend, which performs segmentation, post-processing, feature extraction, beat classification, and final diagnosis.

Once the pipeline completes, the page displays:

- **The normalized ECG signal**, plotted with its P, QRS, and T segments color-coded (Figure 2.11).

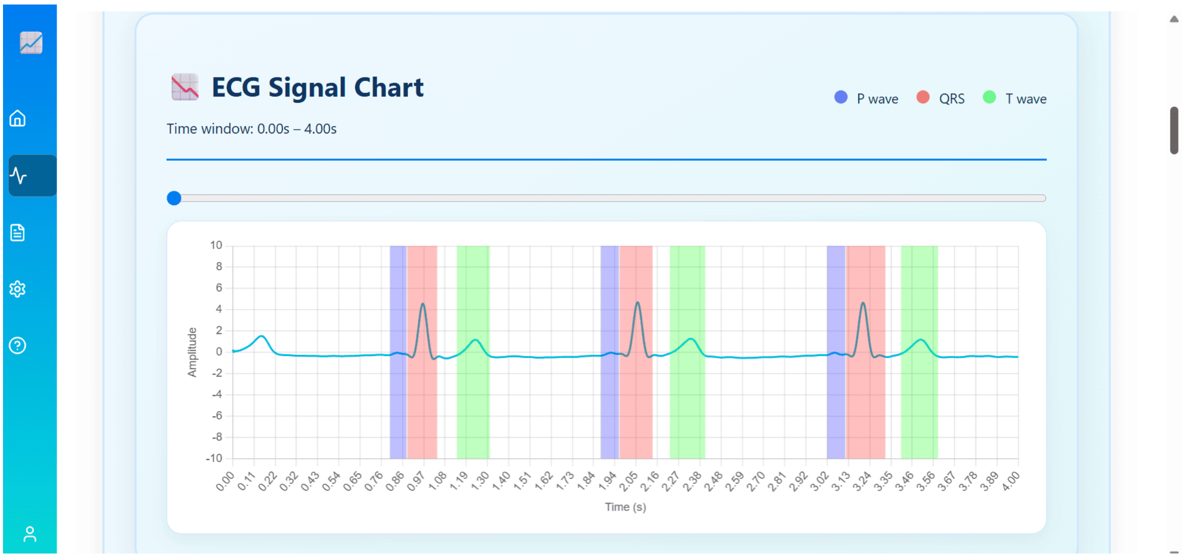


Figure 2.11: Visualization of ECG signal with color-coded wave segmentation (P, QRS, T).

- **A feature viewer**, displaying beat-wise metrics such as QRS duration, QT interval, heart rate, and beat type using graphical representations (Figure 2.12).

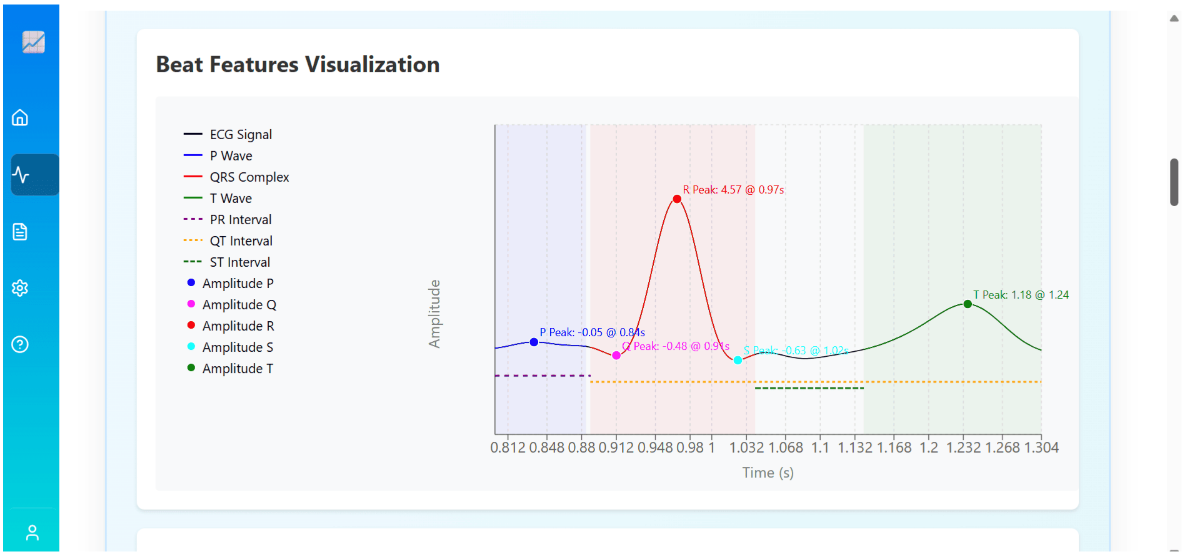


Figure 2.12: Feature viewer showing beat-wise ECG metrics extracted from segmentation.

- **A diagnosis card**, showing whether the result is Normal, Abnormal, or a specific pathology (e.g., Sinus Bradycardia) (Figure 2.13).

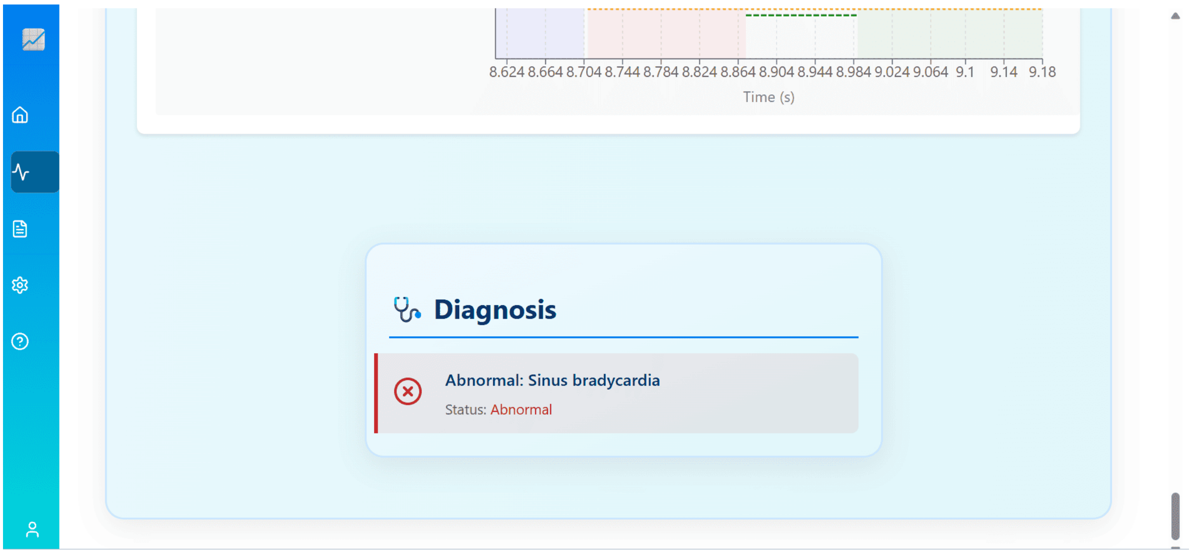


Figure 2.13: Diagnosis card summarizing model decision based on extracted features.

This page reflects the full integration between the frontend and backend, and is dynamically updated based on the user's input and the model predictions.

b Other Pages

While the interface includes a sidebar for navigation (Dashboard, Records, Settings, etc.), some pages — such as the Dashboard — currently display static placeholder data (e.g., total patients, average cost). These are not yet connected to backend analytics, and are reserved for future extensions.

Conclusion

This chapter presented the core of our work: the design and development of an intelligent system for real-time ECG interpretation. By using a multi-agent architecture, we divided the task into manageable steps —from acquisition and segmentation to classification— making the system easier to understand, maintain, and scale.

Through a combination of deep learning and classical machine learning models, we achieved both strong performance and interpretation. The use of the SPADe framework made agent coordination flexible and reliable, supporting real -time processing and modular design.

Overall, the system proves that it is possible to create an automatic ECG interpretation tool which is both effective and practical. The next section will reflect on the overall contribution of the project and discuss future instructions.

General Conclusion

Synthesis

This thesis discussed the implementation of a real-time, intelligent ECG interpretation system. The main goals were to address limitations of conventional ECG analysis systems– lack of modularity, limited real-time ability, lacking interpretability in clinical settings.

In order to address these problems, we developed a multi-agent model, where for each agent, the agents are responsible for signal acquisition, segmentation, feature extraction, and classification. communicate via the SPADe Framework and the XMPP protocol to enable decentralized coordination and scaling of the system.

We adopted a hybrid modelling approach that mixed deep learning and classical machine learning: a 1D U-Net that allow for accurate wave segmentation, followed for detection of R peaks by CNNs, and for diagnosis at beat-level and then at signal-level by Random Forest classifiers. This is 'a desirable synergy as it enables both high performance and better interpretability — the latter being particularly important in healthcare contexts'.

We validated our system on several public ECG databases and its performance was found to be consistent in detecting and classifying important cardiac events. We also developed a web based user interface, which is intuitive to the user to interact with the system and to visualize the results. Overall, our work illustrates that AI systems formulated as agents have a role to play in the timely and accurate diagnosis of heart disease.

Perspectives

Although the present system is based on a solid foundation, there are many avenues of promising work. The following key points present a road-map describing how it can be further developed and acclimatized to the clinical setting.

- **Expansion of Diagnostic Scope:** The most direct path in future work is to extend the diagnostic range of the system. This would include training and incorporating additional binary classifiers for a larger subset of common and serious arrhythmias including, but not limited to AF, PVCs, and types of heart block.
- **Enhancement of Core Models:** There is a rich playground for tuning the performance of the core model. More sophisticated deep learning based architectures, like Transformers or attention-based U-Nets can be considered for the segmentation task in future implementations. Moreover, even ensembling of the pretrained models could be beneficial to get better accuracy and robustness for all models after extensive hyper-parameter optimization and more advanced data augmentation.
- **Integration of Advanced Explainability (XAI):** To further enhance clinical trust and usability, future research efforts should explore integrating advanced XAI methods. By using random forest, it helps to a certain extent to interpret results, but including methods like SHAP (SHapley Additive exPlanations) to provide visualization on which characteristic features or beat morphologies are informative for a certain diagnosis would be a very interesting extension to the user interface.
- **Real-World Clinical Validation:** The ultimate next step is to validate the system under real-world clinical conditions. This would entail performing a pilot study to evaluate its performance on data from local patient populations, testing its real-time WebSocket capabilities in a live monitoring application, and formally comparing its diagnostic outputs against those from certified cardiologists.

Bibliography

[1] Nabila Abraham and Naimul Mefraz Khan. A novel focal tversky loss function with improved attention u-net for lesion segmentation. *arXiv preprint arXiv:1810.07842*, 2019. URL: `https://arxiv.org/abs/1810.07842`.

[2] P.S. Addison. Wavelet transforms and the ecg: A review. *Physiological Measurement*, 26(5), 2005. `doi:10.1088/0967-3334/26/5/R01`.

[3] Yaqoob Ansari et al. Deep learning for ecg arrhythmia detection and classification: An overview of progress for period 2017–2023. *Frontiers in Physiology*, 14, 2023. Available at: `https://www.frontiersin.org/articles/10.3389/fphys.2023.1246746/full`. `doi:10.3389/fphys.2023.1246746`.

[4] Yehuelashet M. Ayano, Friedrich Schwenker, Bisrat D. Dufera, and Taye G. Debelee. Interpretable machine learning techniques in ecg-based heart disease classification: A systematic review. *Diagnostics*, 13(1), 2023. Available at: `https://www.mdpi.com/2075-4418/13/1/111`. `doi:10.3390/diagnostics13010111`.

[5] H. Azami and J. Escudero. Combination of signal segmentation approaches using fuzzy decision making. *Proceedings of the IEEE EMBC*, 2015. Milan, 2015, Available at: `https://www.pure.ed.ac.uk/ws/portalfiles/portal/21442359/EMBC15_1541_FI.pdf`.

[6] S. Bai, J.Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. Available at: `https://arxiv.org/abs/1803.01271`.

[7] Cleveland Clinic. 4 heart valves: What they are and how they work, 2023. URL: `https://my.clevelandclinic.org/health/body/17067-heart-valves`.

[8] Cleveland Clinic. Heart conduction system (cardiac conduction), 2023. Retrieved from https://my.clevelandclinic.org/health/body/21648-heart-conduction-system. URL: https://my.clevelandclinic.org/health/body/21648-heart-conduction-system.

[9] Cleveland Clinic. How blood flows through the heart & body, 2023. URL: https://my.clevelandclinic.org/health/articles/17060-how-does-the-blood-flow-through-your-heart.

[10] G. D. Clifford, C. Liu, B. Moody, L. H. Lehman, I. Silva, Q. Li, A. E. Johnson, and R. G. Mark. Af classification from a short single lead ecg recording: the physionet/-computing in cardiology challenge 2017. *2017 Computing in Cardiology (CinC)*, 2017. doi:10.22489/CinC.2017.065-469.

[11] G.D. Clifford, F. Azuaje, and P. McSharry. Advanced methods and tools for ecg data analysis. *Artech House Biomedical Engineering Series*, 2006. Available at: https://www.academia.edu/99218250/Advanced_methods_and_tools_for_ECG_data_analysis.

[12] Zahra Ebrahimi et al. A review on deep learning methods for ecg arrhythmia classification. 2020. Available at: https://www.researchgate.net/publication/342340435_A_Review_on_Deep_Learning_Methods_for_ECG_Arrhythmia_Classification.

[13] ECGWaves. Normal ecg: P wave, qrs complex, st segment, t wave, j point, 2023. URL: https://ecgwaves.com/topic/ecg-normal-p-wave-qrs-complex-st-segment-t-wave-j-point/.

[14] Elsan. Électrocardiogramme (ecg), 2023. URL: https://www.elsan.care/fr/pathologie-et-traitement/maladies-cardiovasculaires/electrocardiogramme-ECG-definition-deroulement-resultats.

[15] EZmed Learning. Conduction system of the heart, 2023. URL: https://www.ezmedlearning.com/blog/conduction.

[16] S. Ghimire et al. Semantic segmentation of 12-lead ecg using 1d residual u-net with squeeze-excitation blocks. *Applied Sciences*, 12(7), 2022. Available at: https://www.mdpi.com/2076-3417/12/7/3332. doi:10.3390/app12073332.

[17] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley. Mit-bih normal sinus rhythm database. PhysioNet, 2000. doi:10.13026/C2NK5R.

[18] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000. doi:10.1161/01.CIR.101.23.e215.

[19] P.S. Hamilton and W.J. Tompkins. Quantitative investigation of qrs detection rules using the mit/bih arrhythmia database. *IEEE Transactions on Biomedical Engineering*, BME-33(12), 1986. Available at: https://ieeexplore.ieee.org/document/4122227.

[20] Awni Y. Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, et al. Cardiologist-level arrhythmia detection with deep neural networks. *Nature Medicine*, 25:65–69, 2019. Retrieved from https://www.nature.com/articles/s41591-018-0268-3. doi:10.1038/s41591-018-0268-3.

[21] A.Y. Hannun, P. Rajpurkar, M. Haghpanahi, et al. Cardiologist-level arrhythmia detection with deep neural networks. *Nature Medicine*, 25, 2019. Available at: https://www.nature.com/articles/s41591-018-0268-3. doi:10.1038/s41591-018-0268-3.

[22] O. Hussein, J.M.H. Altmemi, M.A. Abbas, A. Uğurver, et al. Improving automated labeling with deep learning and signal segmentation for accurate ecg signal analysis. *Service Oriented Computing and Applications*, 2024. doi:10.1007/s11761-024-00436-5.

[23] S. Iqbal, W. Altaf, M. Aslam, W. Mahmood, and M.U.G. Khan. Application of intelligent agents in health-care: Review. *Artificial Intelligence Review*, 46(1):83–112, 2016. doi:10.1007/s10462-016-9457-y.

[24] Komal S. Jaisinghani and Sandeep Malik. Ecg-based heartbeat classification using machine learning: Survey. *BBRC*, 2021. Available at: https://bbrc.in/wp-content/uploads/2021/01/13_14-SPL-Galley-proof-094.pdf.

[25] C. Joung et al. Deep learning based ecg segmentation for delineation of diverse arrhythmias. *arXiv preprint arXiv:2304.06237*, 2023. Available at: https://arxiv.org/abs/2304.06237.

[26] Journal of Medical Engineering & Technology. Ecg denoising and feature extraction techniques – a review. 2021. Available at: https://www.tandfonline.com/doi/abs/10.1080/03091902.2021.1955032.

[27] Yusipov I. Moskalenko V. Nikolskiy A. Kosonogov K. Zolotykh N. Ivanchenko M. (2021). Kalyakulina, A. Ludb: Lobachevsky university electrocardiography database. 2021. doi:10.13026/eegm-h675.

[28] S. Karpagachelvi, M. Arthanari, and M. Sivakumar. Ecg feature extraction techniques – a survey approach. *arXiv preprint arXiv:1005.0957*, 2010. Available at: https://arxiv.org/abs/1005.0957.

[29] R. Kher. Signal processing techniques for removing noise from ecg signals. *Journal of Biomedical Engineering and Research*, 2019. doi:10.17303/JBER.2019.3.101.

[30] Y. Kilic et al. Explainable arrhythmia detection using shap and deep neural networks. *Journal of Healthcare Engineering*, 2025. Available at: https://link.springer.com/article/10.1007/s40846-025-00949-0. doi:10.1007/s40846-025-00949-0.

[31] P. Laguna, R. G. Mark, A. L. Goldberger, and G. B. Moody. A database for evaluation of algorithms for measurement of qt and other waveform intervals in the ecg. In *Computers in Cardiology 1997*, pages 673–676, 1997. doi:10.1109/CIC.1997.648140.

[32] S. Li and G. Zhao. P wave detection in electrocardiogram based on wavelet transform and differential correction. *ResearchGate*, 2020. Available at: https://www.researchgate.net/publication/343900111.

[33] X. Li et al. Ecg classification using a transformer-based framework. *Computers in Biology and Medicine*, 2022. doi:10.1016/j.compbiomed.2022.105325.

[34] A. Mahmood, N.A. Khan, and S. Mohsin. Performance analysis of windowing techniques for ecg signal feature extraction. *International Journal of Computer Applications*, 2012.

[35] Seyed Matin Malakouti. Heart disease classification based on ecg using machine learning models. *Biomedical Signal Processing and Control*, 2023. doi:10.1016/j.bspc.2023.104796.

[36] M.S. Manikandan and S. Dandapat. An electrocardiogram signal compression techniques: a comprehensive review. *Analog Integrated Circuits and Signal Processing*, 97:425–437, 2018. Available at: https://link.springer.com/article/10.1007/s10470-018-1323-1. doi:10.1007/s10470-018-1323-1.

[37] S.S. Mehta and N.S. Lingayat. Detection of p and t-waves in electrocardiogram using support vector machine. *Proceedings of the WCECS*, 2008. San Francisco, 2008, availble at: https://www.researchgate.net/publication/44262470.

[38] G. B. Moody and R. G. Mark. The impact of the mit-bih arrhythmia database. volume 20, pages 45–50, 2001. doi:10.1109/51.932724.

[39] V. Moskalenko, E. Andreozzi, D. Esposito, and G. Gargiulo. Deep learning for ecg segmentation. *arXiv preprint arXiv:2001.04689*, 2020. Available at: https://arxiv.org/abs/2001.04689.

[40] Abhishek Ojha, Prashant Verma, and Rajesh Srivastava. Explaining deep learning-based ecg arrhythmia classification using lime and shap. *ACM Transactions on Computing for Healthcare*, 2024. Available at: https://dl.acm.org/doi/fullHtml/10.1145/3643488.3660294. doi:10.1145/3643488.3660294.

[41] J. Palanca, A. Terrasa, V. Julián, and C. Carrascosa. Spade 3: Supporting the new generation of multi-agent systems. *IEEE Access*, 8:182537–182549, 2020. URL: https://ieeexplore.ieee.org/document/9207929, doi:10.1109/ACCESS.2020.3029141.

[42] J. Pan and W.J. Tompkins. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32(3):230–236, 1985. Available at: https://ieeexplore.ieee.org/document/4122029.

[43] Ahmad Luqman Md Pauzi et al. Deep learning-based ecg arrhythmia classification: A systematic review. *Applied Sciences*, 13(8), 2023. Available at: https://www.mdpi.com/2076-3417/13/8/4964. doi:10.3390/app13084964.

[44] Harshadkumar B. Prajapati, S.H. Jambukia, and V.K. Dabhi. Classification of ecg signals using machine learning techniques: A survey. *Proceedings of ICACEA*, pages 714–719, 2015. Available at: https://www.researchgate.net/publication/280462158.

[45] PubMed. Improved t-wave detection in ecg signals based on non-decimated stationary wavelet transform. *PubMed*, 2023. Available at: https://pubmed.ncbi.nlm.nih.gov/37944176/.

[46] M. A. Reyna, N. Sadr, E. A. Pérez-Alday, A. Gu, A. J. Shah, C. Robichaux, A. I. Wong, C. Liu, A. Elola, S. Seyedi, et al. Will two do? varying dimensions in electrocardiography: The physionet/computing in cardiology challenge 2021. *Physiological Measurement*, 2021. doi:10.23919/CinC53138.2021.9662687.

[47] J. A. Rincón, C. Carrascosa, and V. Julián. Flexible agent architecture mixing reactive and deliberative behaviors in spade. *Preprint*, 2023. URL: https://www.researchgate.net/publication/367551179.

[48] Peter Saint-Andre. Extensible messaging and presence protocol (xmpp): Core. https://xmpp.org/rfcs/rfc6120.html, 2011. RFC 6120.

[49] Yasar Sattar and Lovely Chhabra. Electrocardiogram, 2025. Available from: https://www.ncbi.nlm.nih.gov/books/NBK549803/. URL: https://www.ncbi.nlm.nih.gov/books/NBK549803/.

[50] Charles Severance. Discovering javascript object notation. *IEEE Computer*, 45(4), 2012. URL: https://www.researchgate.net/publication/241638061_Discovering_JavaScript_Object_Notation.

[51] E. Shakshuki and M. Reid. Multi-agent system applications in healthcare: Current technology and future roadmap. *Procedia Computer Science*, 52:252–261, 2015. doi:10.1016/j.procs.2015.05.071.

[52] A.K. Singh and S. Krishnan. Ecg signal feature extraction trends in methods and applications. *BioMedical Engineering OnLine*, 22, 2023. Available at: https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/s12938-023-01075-1. doi:10.1186/s12938-023-01075-1.

[53] StatPearls. Anatomy, thorax, heart, 2023. Retrieved from https://www.ncbi.nlm.nih.gov/books/NBK470256/. URL: https://www.ncbi.nlm.nih.gov/books/NBK470256/.

[54] L. Sörnmo and P. Laguna. Electrocardiogram (ecg) signal processing. *Encyclopedia of Biomedical Engineering*, 2006. doi:10.1002/9780471740360.ebs1482.

[55] A. Taddei, G. Distante, M. Emdin, P. Pisani, G. B. Moody, P. Zecchi, and C. Marchesi. The european st-t database: standard for evaluating systems for the analysis of st-t changes in ambulatory electrocardiography. In *European Heart Journal*, volume 13, pages 1164–1172, 1992. doi:10.1093/oxfordjournals.eurheartj.a060332.

[56] Texas Heart Institute. The coronary arteries, 2023. URL: https://www.texasheart.org/heart-health/heart-information-center/topics/the-coronary-arteries/.

[57] Texas Heart Institute. Electrocardiogram (ecg or ekg), 2023. Retrieved from https://www.texasheart.org/heart-health/heart-information-center/topics/electrocardiogram/. URL: https://www.texasheart.org/heart-health/heart-information-center/topics/electrocardiogram/.

[58] Texas Heart Institute. Heart anatomy, 2023. URL: https://www.texasheart.org/heart-health/heart-information-center/topics/heart-anatomy/.

[59] VectorStock. Human heart structure diagram – medical science vector illustration, 2023. URL: https://www.vectorstock.com/royalty-free-vector/human-heart-structure-diagram-medical-science-vector-47854656.

[60] Prashant Verma. Explainable deep learning approaches in ecg-based rhythm classification. *arXiv preprint arXiv:2205.02803*, 2022. Available at: https://arxiv.org/abs/2205.02803.

[61] VIDAL. Électrocardiogramme (ecg), 2023. URL: https://www.vidal.fr/sante/examens-tests-analyses-medicales/electrocardiogramme-ecg-holter.html.

[62] World Health Organization. Cardiovascular diseases (cvds), 2023. URL: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

[63] Michael Yeung, Evis Sala, Carola-Bibiane Schönlieb, and Leonardo Rundo. Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *arXiv preprint arXiv:2102.04525*, 2021. URL: https://arxiv.org/abs/2102.04525.

[64] O. Yildirim, U.B. Baloglu, R.S. Tan, and U.R. Acharya. A 1d convolutional neural network model for ecg signal classification. *Computers in Biology and Medicine*, 103, 2018. doi:10.1016/j.compbiomed.2018.05.013.

[65] Guo H. Chu H Zheng, J. A large scale 12-lead electrocardiogram database for arrhythmia study. 2022. doi:10.13026/wgex-er52.

[66] A. Zifan, Y. Zhang, and M. Karim. Automated ecg segmentation using piecewise derivative dynamic time warping. *International Journal of Biological and Medical Sciences*, 1(3), 2006. Available at: https://www.researchgate.net/publication/228734344.

Acronyms

AI Artificial Intelligence

API Application Programming Interface

BPM Beats Per Minute

CNN Convolutional Neural Network

CVD Cardiovascular Disease

ECG Electrocardiogram

EDB European ST-T Database

IoU Intersection over Union

JSON JavaScript Object Notation

LBBB Left Bundle Branch Block

LSTM Long Short-Term Memory

LUDB Lobachevsky University Electrocardiography Database

MAS Multi-Agent System

MITDB Massachusetts Institute of Technology - Beth Israel Hospital Database

ML Machine Learning

NTIC New Technologies of Information and Communication

NSRDB Normal Sinus Rhythm Database

PTBDB Physikalisch-Technische Bundesanstalt Database

QTDB QT Database

RBBB Right Bundle Branch Block

REST Representational State Transfer

RF Random Forest

RMSSD Root Mean Square of Successive Differences

RNN Recurrent Neural Network

SPADE Smart Python Agent Development Environment

SVM Support Vector Machine

TCN Temporal Convolutional Network

UI User Interface

UML Unified Modeling Language

XAI Explainable Artificial Intelligence

XMPP Extensible Messaging and Presence Protocol