Low Complexity Algorithms for Arrhythmia Detection from ECG Signals

Presented by:
Group 20
Isuri Devindi (E/17/058)
Sashini Liyanage (E/17/190)

Supervisors:
Prof. Roshan Ragel
Dr. Titus Jayarathne

17.9 million people died from CVDs in 2019

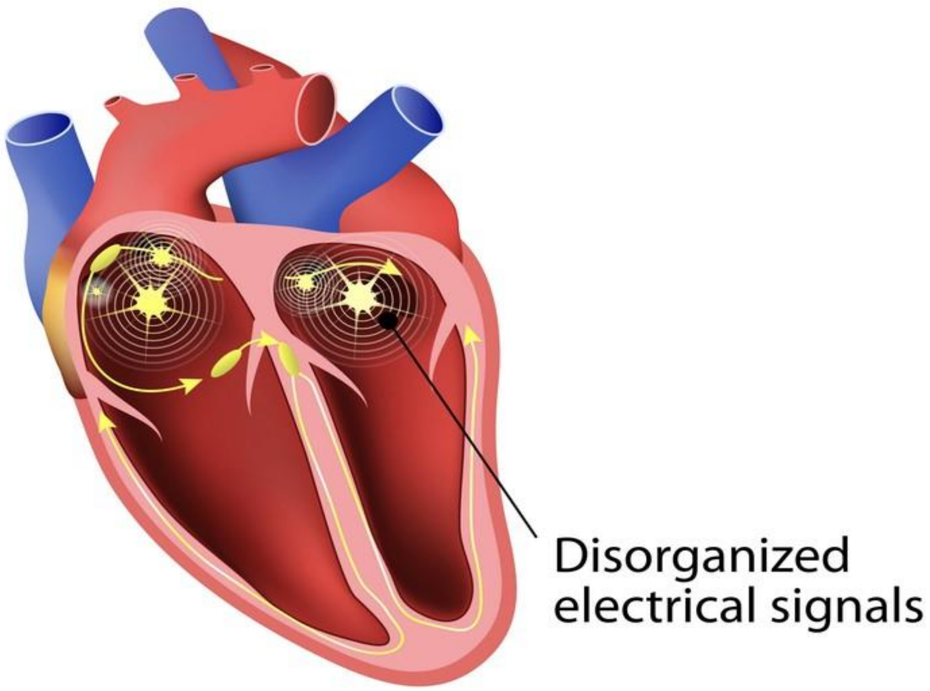
32% of all global deaths

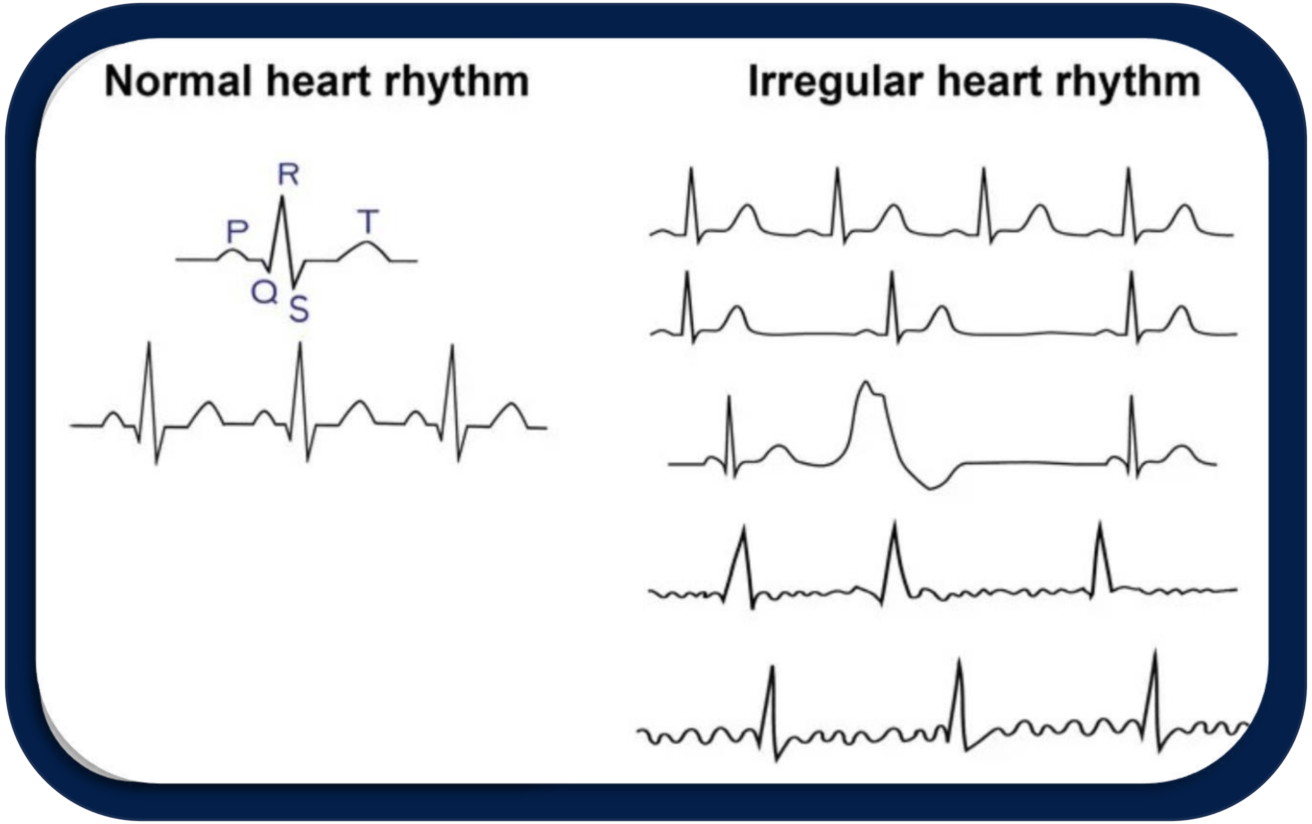
85% were due to heart attack and stroke

Increased to 20.5 million in 2021

Arrhythmia

An arrhythmia is an abnormal heart rhythm





Arrhythmia Detection

Evolved from **stationary** ECG monitoring systems to **wearable** devices

Current methods are not addressing the null power consumption environment

There are no off-the-shelf packages that can be deployed on microcontrollers

Low complexity Real-time Arrhythmia Detection Software

Pre-packaged arrhythmia detection software

Low-complexity algorithm to detect different arrhythmia conditions from ECG signals

Compatible with resource-limited microcontrollers

Main Tasks in an Arrhythmia Classification System

- R-peak Detection
- Segmentation based on R-peaks
- Suspected beat identification
- Compression and transmission of suspected beats
- Improved classification off the edge device

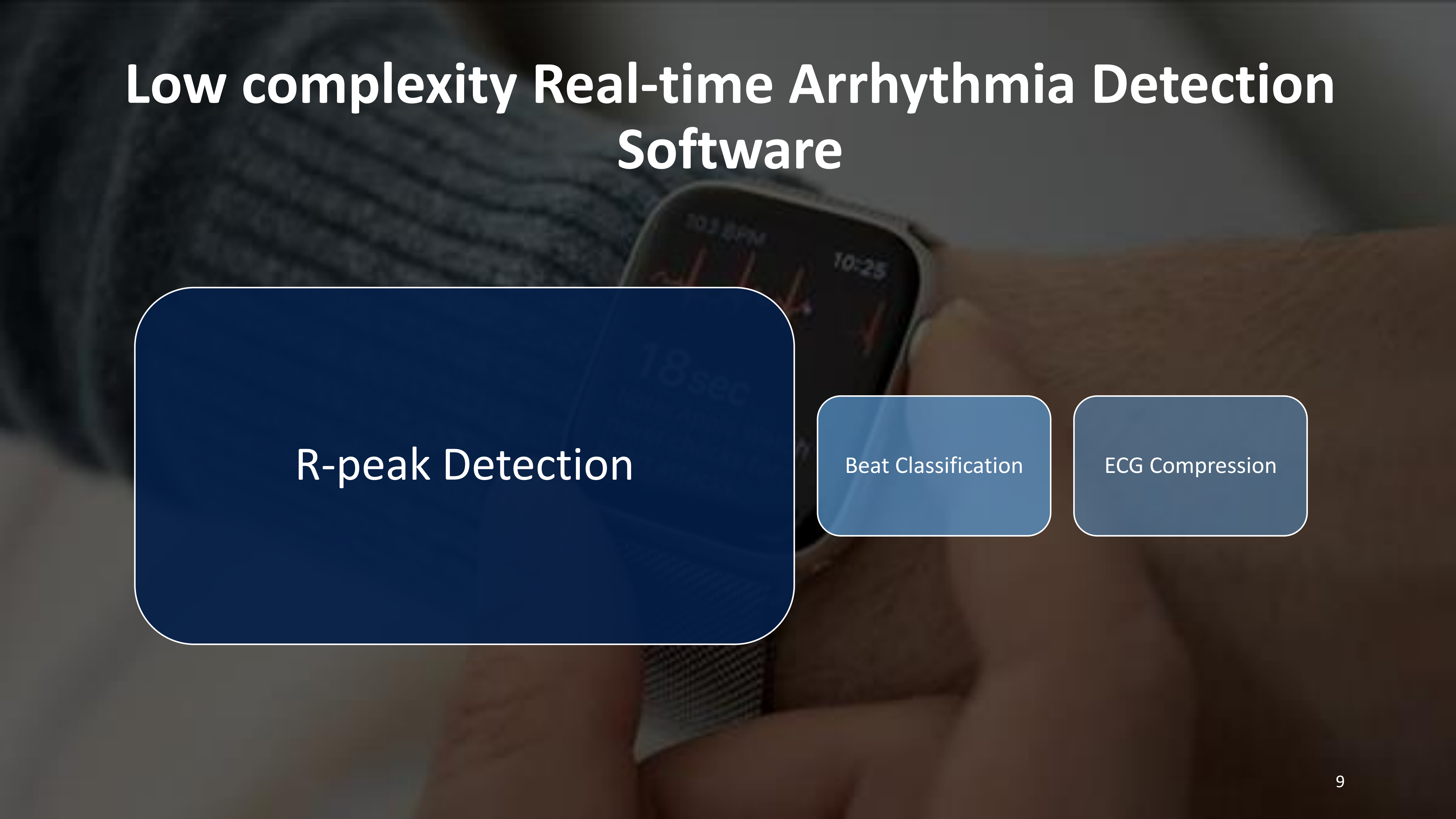
Low complexity Real-time Arrhythmia Detection Software

R-peak Detection

Beat Classification

ECG Compression

Low complexity Real-time Arrhythmia Detection Software



R-peak Detection

Beat Classification

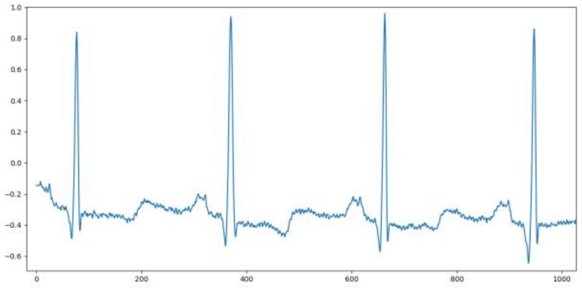
ECG Compression

Experiments: R-peak Detection

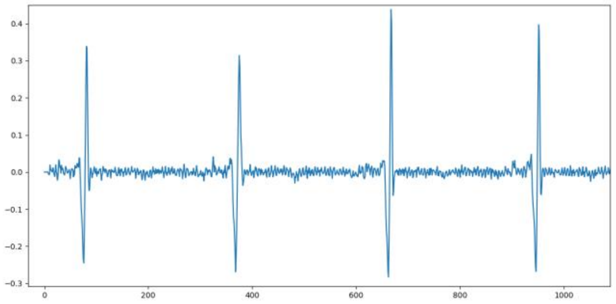
Rule Based R-peak Detection

Leaky Boundary Based R-peak Detection

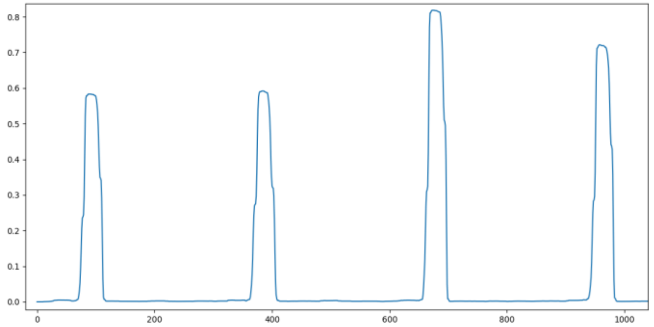
Experiment: Rule Based R-peak Detection



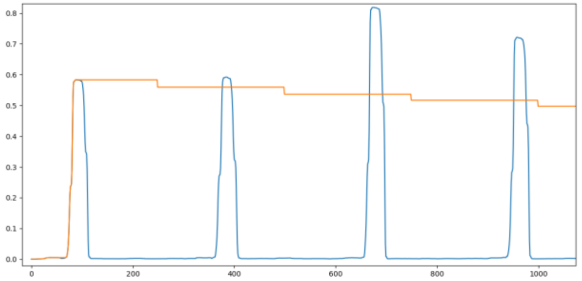
ECG Signal



High-pass filter
(Moving average)



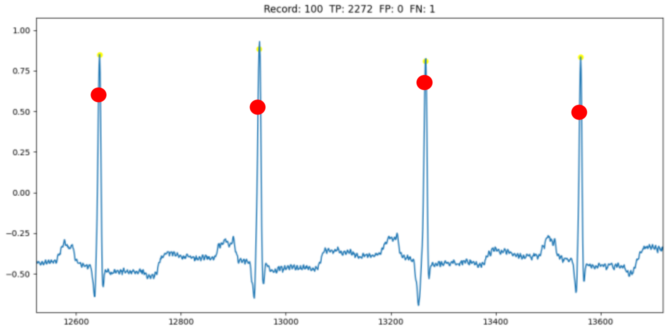
Squaring and moving window summation



Thresholding

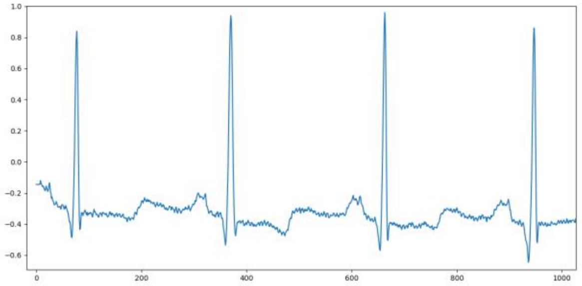
Rule base

- Height
- Heart-rate

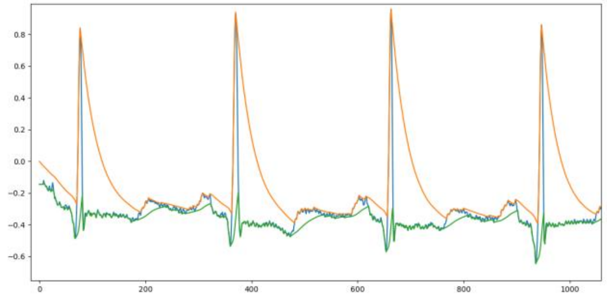


Detected R-Peaks

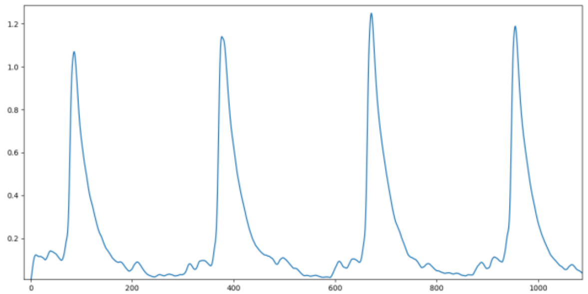
Experiment: Leaky Boundary-Based R-peak Detection



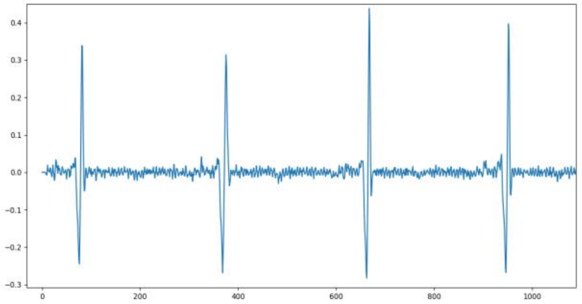
ECG Signal



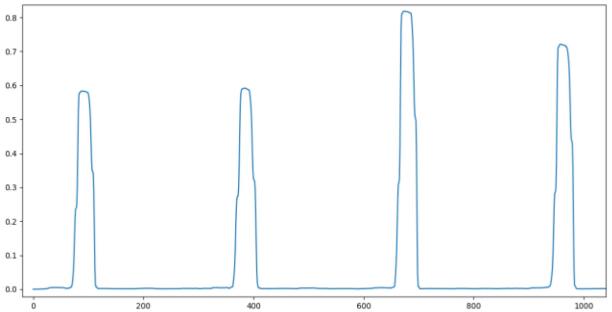
Boundary of the ECG



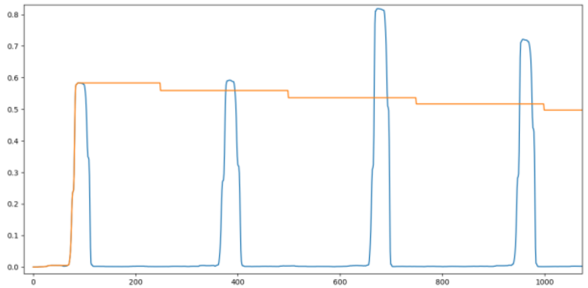
Low-pass filter to the
boundary difference



High pass filter
(Moving average)



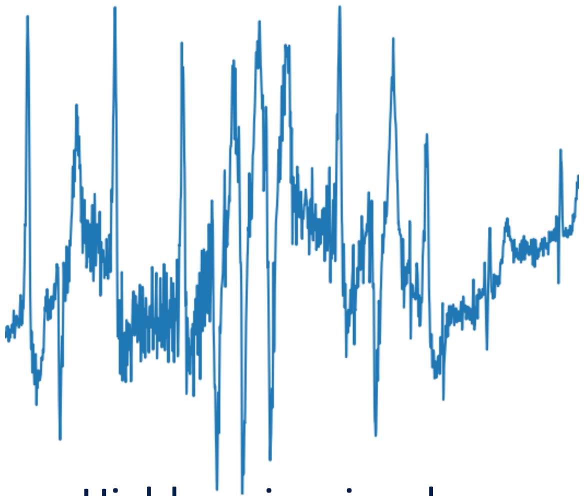
Squaring and moving
window summation



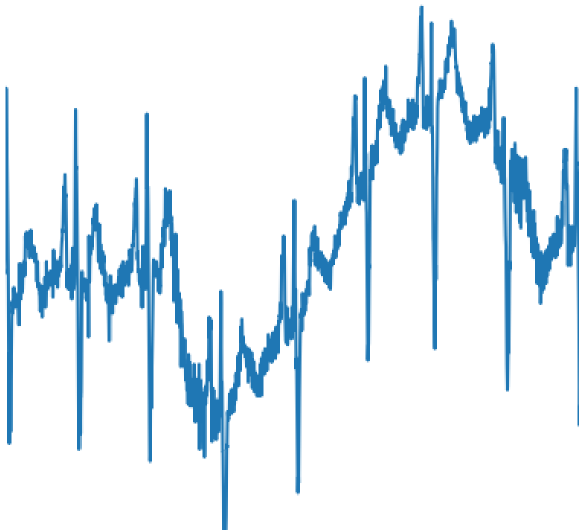
Thresholding

Findings: R-peak Detection Result

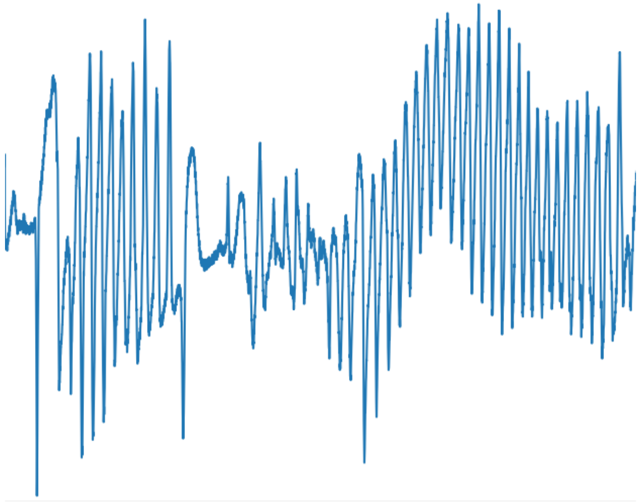
	Sensitivity	Accuracy	Precision
Rule based R-peak detection	99.24	98.23	98.94
Leaky boundary-based R-peak detection	99.64	**99.06**	99.41



Highly noisy signal

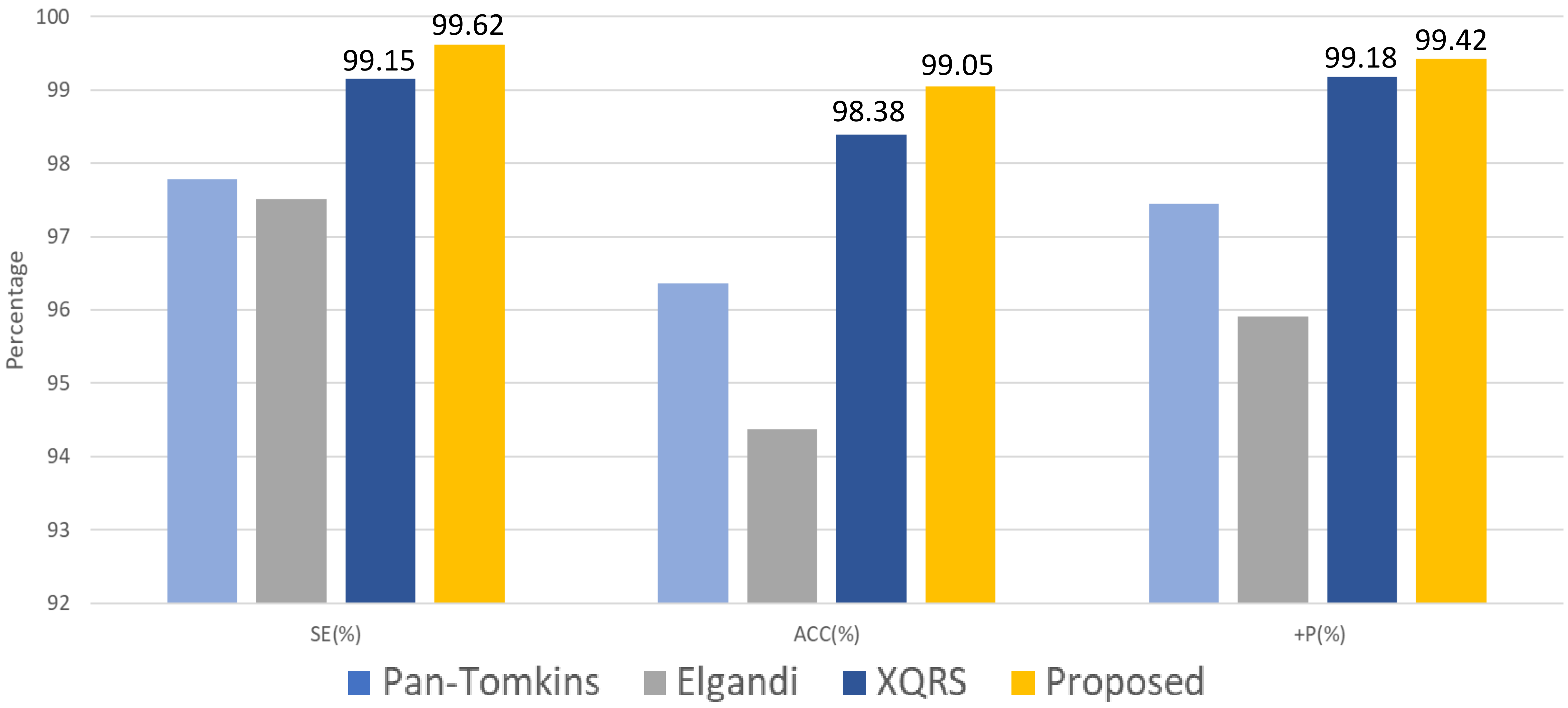


Signal with baseline wander



Pacemakers with no R-peaks

Findings: R-peak Detection Result



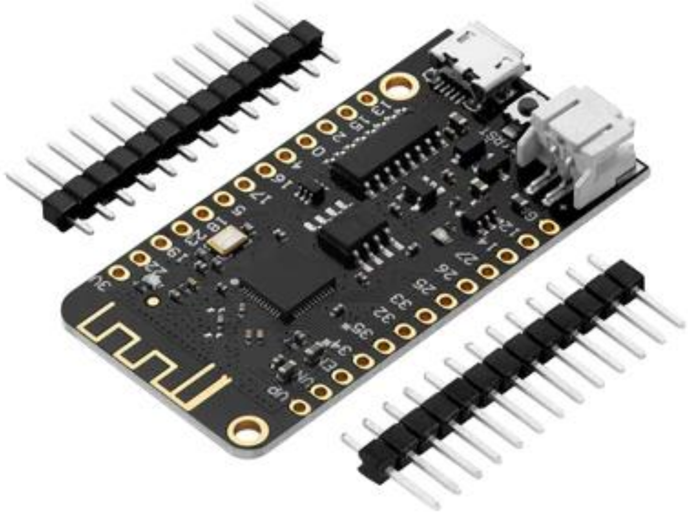
Findings: Testing the Leaky boundary-based R-peak detection algorithm on Hardware

- **Memory utilization:**
  - **RAM: 1.3 KB**

- **Processing time per sample: ~38 nanoseconds**

- Sensor data is serially transmitted to the MCU via UART module, and compression algorithm is run for each sensor data point received

LOLIN32 LITE v1.0.0 with ESP32



SRAM: 520 KB
Flash: 4MB
Max clock frequency: 240 MHz

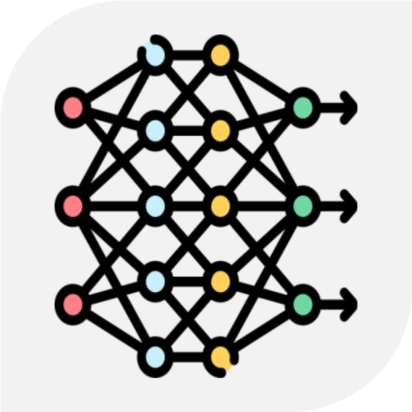
Low complexity Real-time Arrhythmia Detection Software

R-peak Detection

Beat Classification

ECG Compression

Experiments: Beat Classification

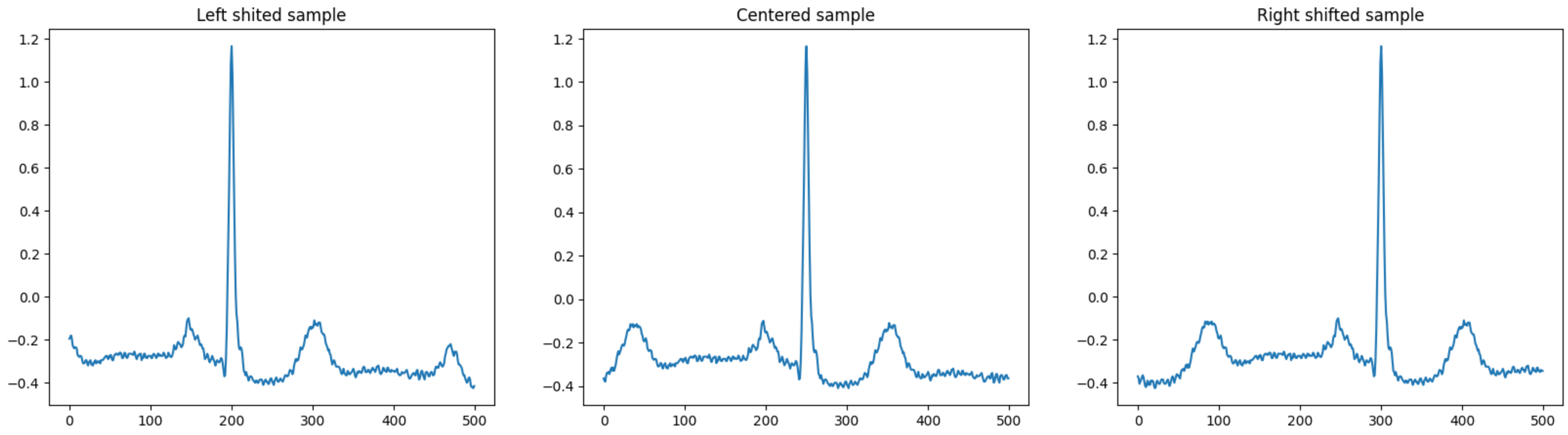


Convolution Neural Network

Spiking Neural Network

Rule Based Classification

Experiment: Dataset Generation for Beat Classification



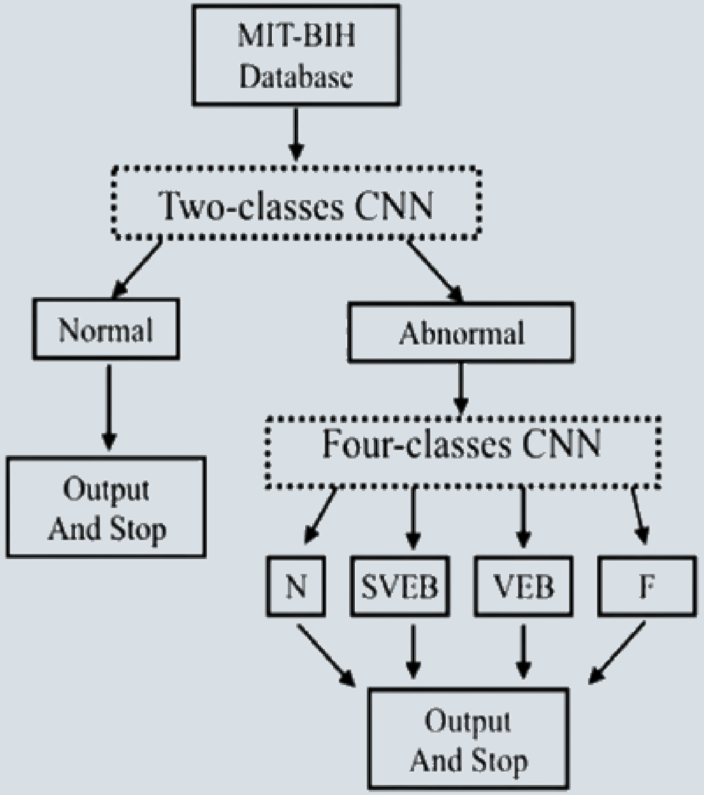
Database: MIT-BIH Arrhythmia Database

Increased the amount of data by generating 3 samples for each annotation timestamp as follows:

- Window with 500 samples centering the R peak (centered).
- Window with 200 from left of the peak and 300 from right of the peak (Left shifted)
- Window with 300 from left of the peak and 200 from right of the peak (Right shifted)

Findings: Optimizing Beat Classification

Power optimization using a two-stage CNN



Two-class CNN (4 layers)	Precision	Recall	F1-Score
Normal	0.95	0.97	0.96
Abnormal	0.72	0.58	0.64

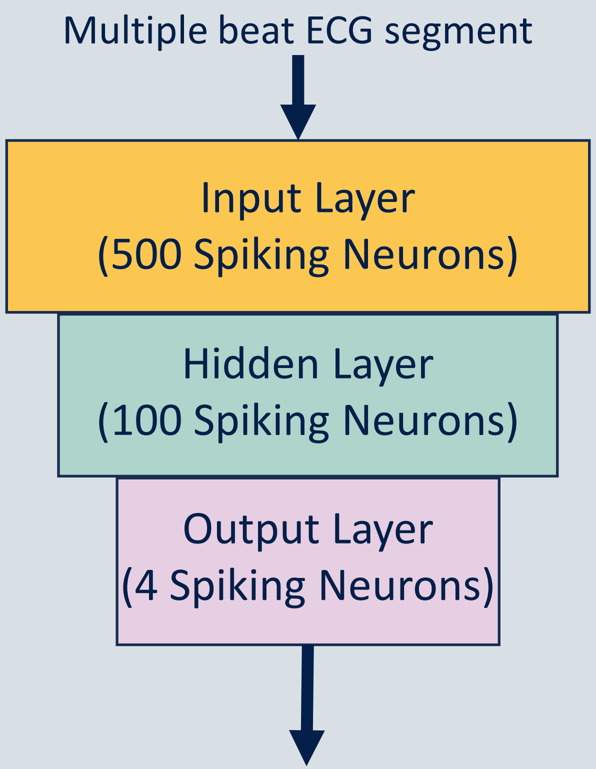
Overall Accuracy 93%

Four-class CNN (18 layers)	Precision	Recall	F1-Score
N	0.96	0.80	0.87
SVEB	0.15	0.19	0.17
VEB	0.64	0.85	0.73
F	0.01	0.21	0.03

Overall Accuracy 80%

Findings: Optimizing Beat Classification

Power optimization using a four-class SNN



Neuron Spiking at highest rate after 20 simulation timesteps taken as the class of the segment

Threshold used for spike generation: 0.01
Neuron threshold: 1.0

Four-class SNN (3 layers)	Precision	Recall	F1-Score
N	0.95	0.83	0.88
SVEB	0.04	0.06	0.05
VEB	0.38	0.81	0.52
F	0.02	0.10	0.04

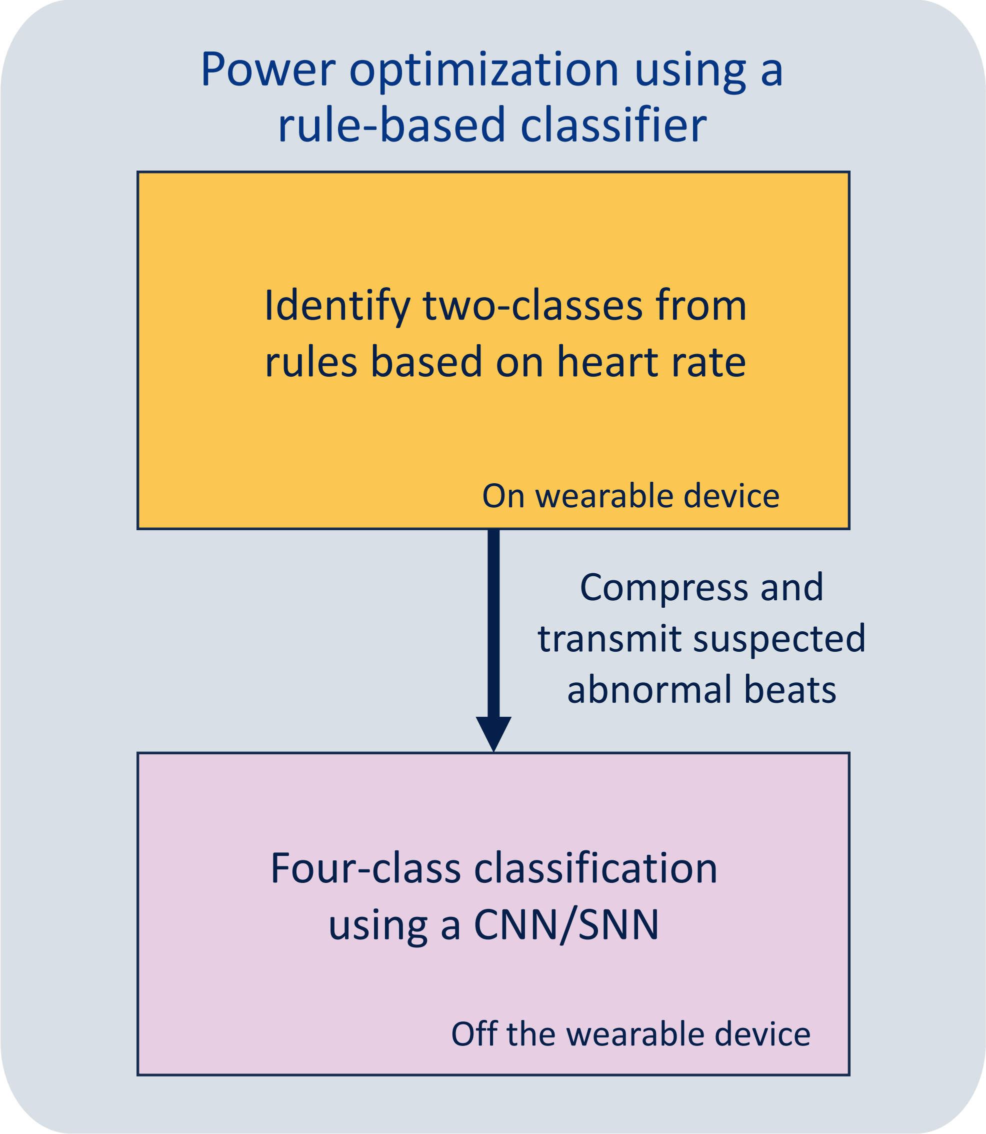
Threshold used for spike generation: 0.06
Neuron threshold: 0.9

Four-class SNN (3 layers)	Precision	Recall	F1-Score
N	0.94	0.92	0.93
SVEB	0.09	0.07	0.08
VEB	0.54	0.80	0.65
F	0.02	0.04	0.02

Overall Accuracy 79%

Overall Accuracy 87%

Findings: Optimizing Beat Classification



Using the standard values for R-R distances;

Rule 1: Normal(N) Vs. Abnormal (SVEB, VEB, F) filtering

If (R-R distance ~ 0.6 to 1.0 seconds) → Normal
else → abnormal

Rule 2: SVEB/F Vs. VEB filtering

If (R-R distance ~ 0.24 to 0.4 seconds) → SVEB or F
else → VEB

On the Edge Device

Rule 3: For SVEB detection

- Rate > 150-250 bpm
- QRS complex is narrow < 120 ms

Rule 4: For VEB detection

- **Abnormal morphology**
- QRS complex is broad > 120 ms
- Premature broad QRS occurs in between normal QRS patterns

Rule 5: For F detection

- **Abnormal morphology**
- QRS complex is a combination of normal and broad QRS

Findings: Finding the optimal threshold that maximizes F1-score

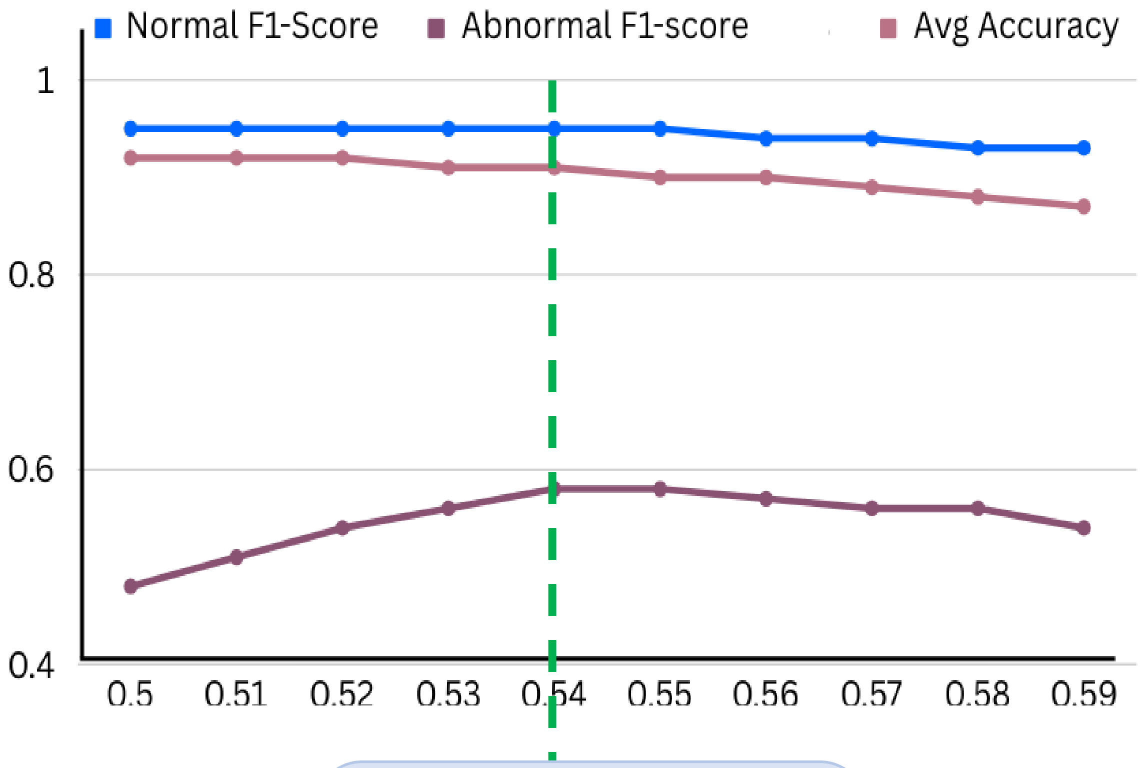
Using the standard values for R-R distances;

Rule : Normal(N) Vs. Abnormal (SVEB, VEB, F) filtering

If (R-R distance < threshold seconds) → Abnormal

else → Normal

On the Edge Device

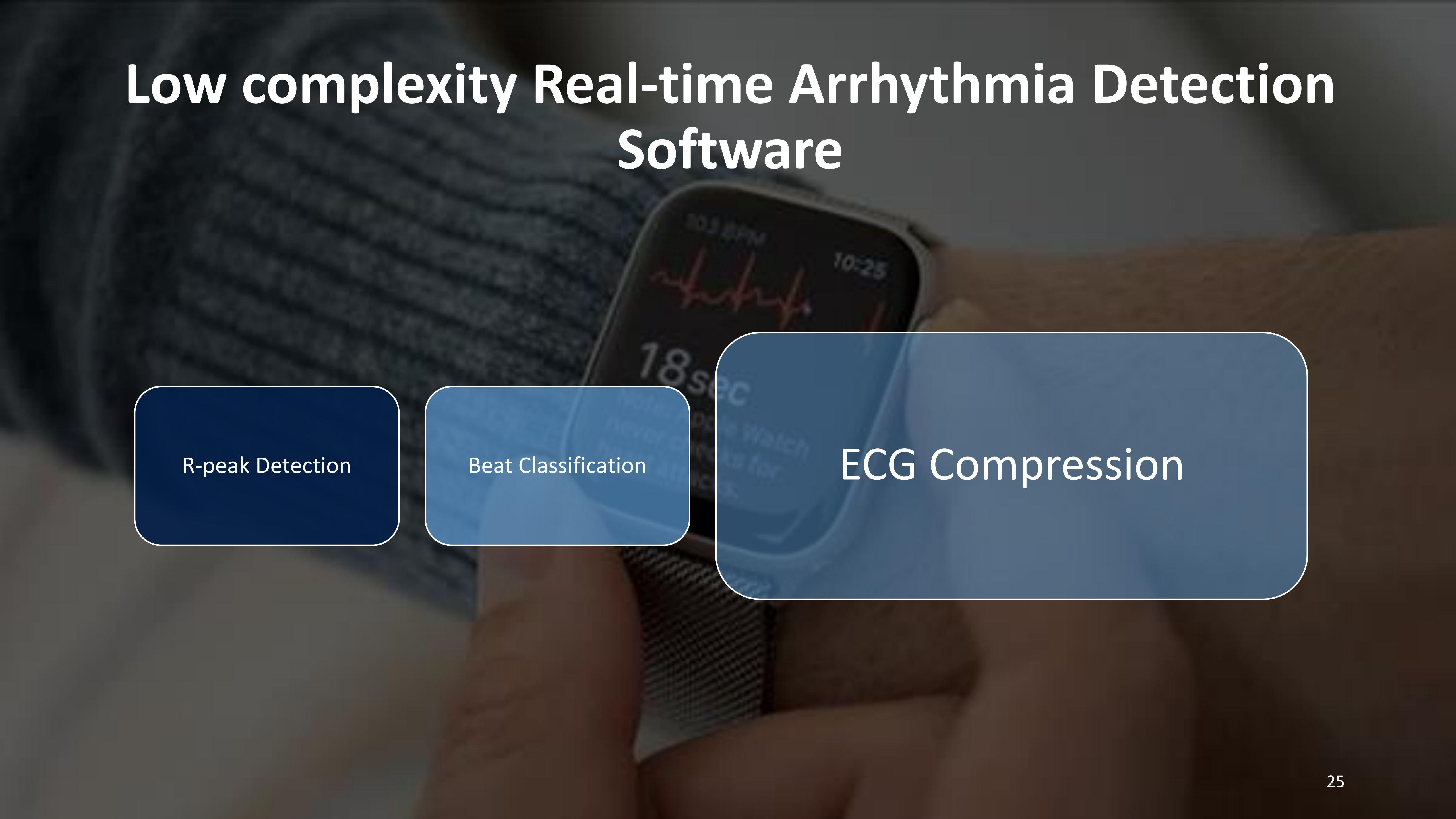


Threshold = 0.54

Normal F1-Score: 0.95

Abnormal F1-Score: 0.58

Average Accuracy: 91%



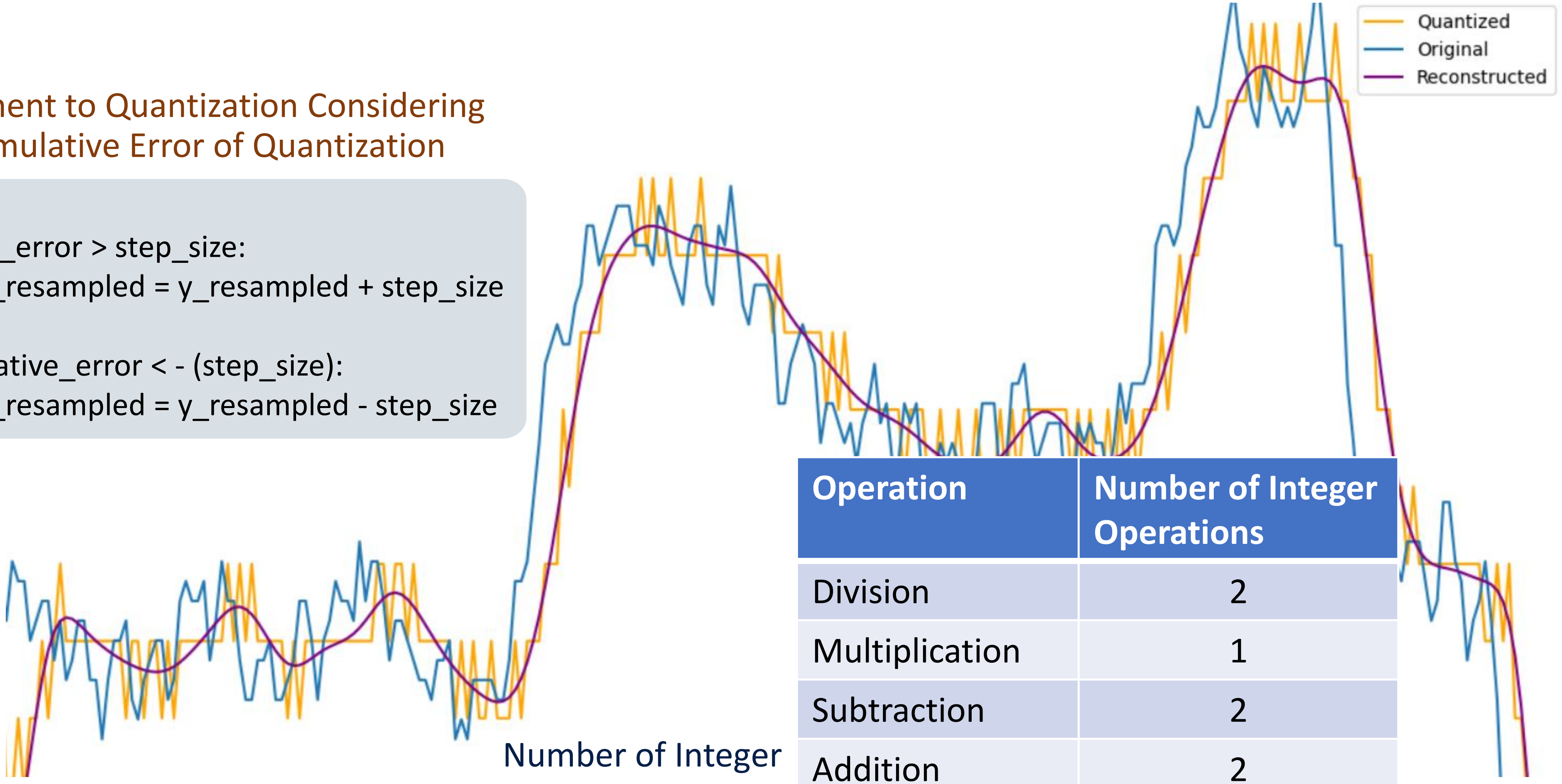
Low complexity Real-time Arrhythmia Detection Software

R-peak Detection

Beat Classification

ECG Compression

Experiment: Hybrid PWM-Quantization ECG Compression



Improvement to Quantization Considering
the Cumulative Error of Quantization

```
If cumulative_error > step_size:
        y_resampled = y_resampled + step_size

else if cumulative_error < - (step_size):
        y_resampled = y_resampled - step_size
```

Number of Integer
Operations Per Sample Point

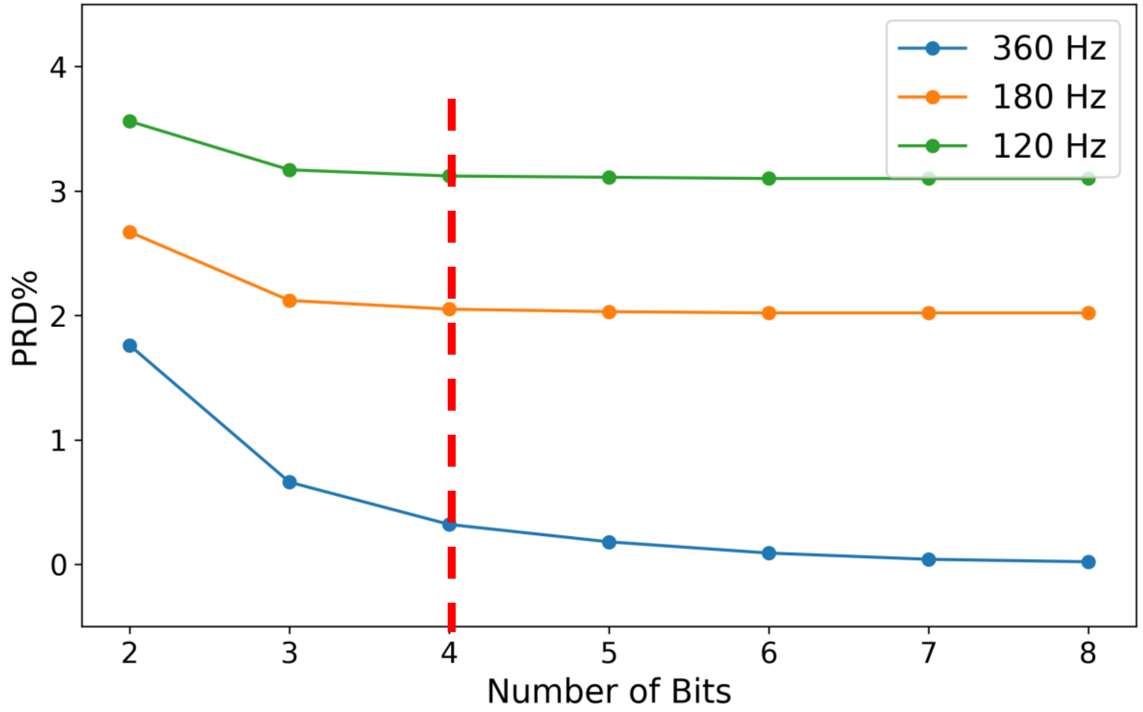
Operation	Number of Integer Operations
Division	2
Multiplication	1
Subtraction	2
Addition	2
Comparison	2
Total	9

Findings: Hybrid PWM-Quantization ECG Compression

360 Hz

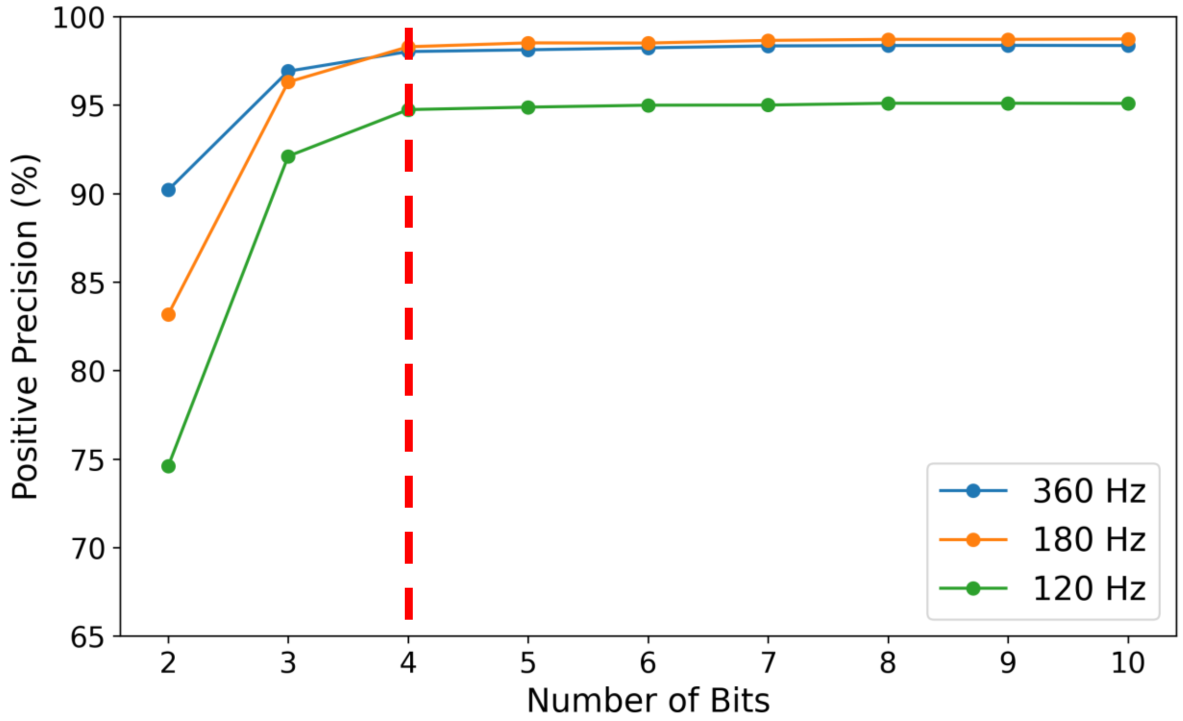
180 Hz

120 Hz



Percentage Root Mean Square Error of Compression Vs Number of Bits

Nearly Perfect Reconstruction (N-PR) for (< 0.5% PDR) for 4 bits (and higher)



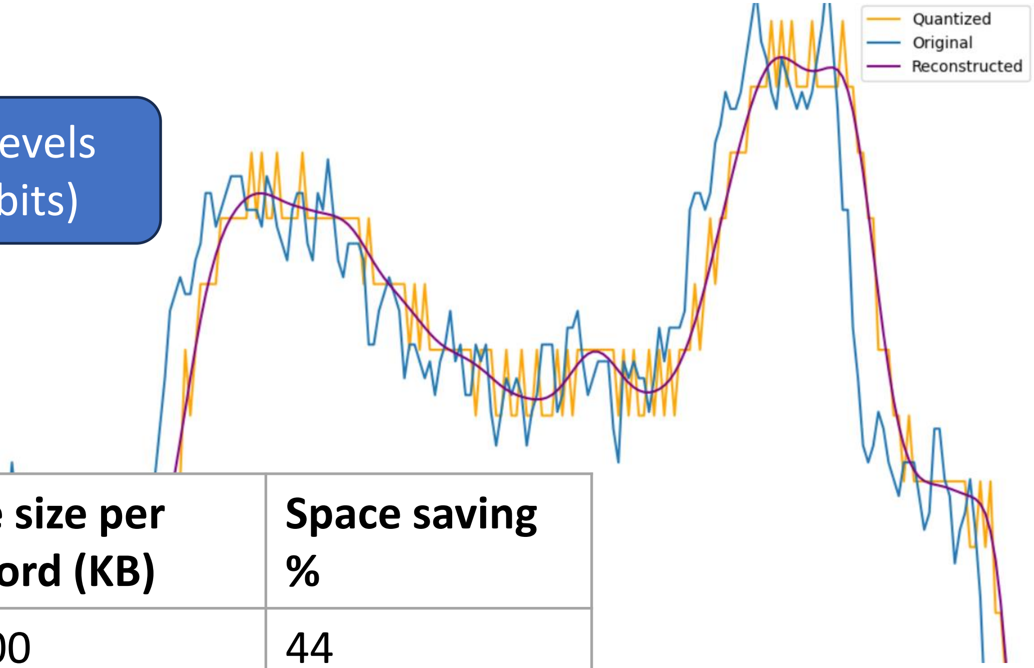
Positive Precision of Pan-Tompkins Algorithm for Reconstructed Signal Vs Number of Bits

Approximately same performance for 4 bits (and higher)

Findings: File size reduction per 30min ECG signal recording

2048 levels (11 bits) → 16 levels (4 bits)

3400 KB



Storing method	**File size per record (KB)**	**Space saving %**
Integers in text files (360Hz)	1900	44
Down-sampled to 180 Hz	900	73
Down-sampled to 120 Hz	600	82
16 sample points as a single 64bit word in binary files (360Hz)	318	91

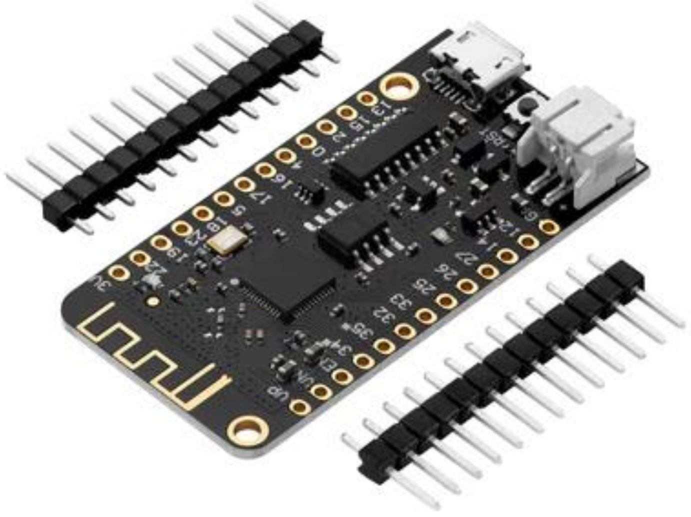
Findings: Testing the compression algorithm on Hardware

- **Memory utilization:**
  - **RAM: 16 bytes**
  - **Flash: 352 bytes**

- **Processing time per sample: ~30 nanoseconds**

- Sensor data is serially transmitted to the MCU via UART module, and compression algorithm is run for each sensor data point received

LOLIN32 LITE v1.0.0
with ESP32



SRAM: 520 KB
Flash: 4MB
Max clock frequency: 240 MHz

Expected Outcomes & Deliverables

R-peak Detection:

- Developed the leaky-boundary based R-peak detection algorithm
  - **Low complex**
  - **Real time**
  - **Packaged as a C program**

Beat Classification:

- Developed an on-edge device rule-based binary classifier
  - **Low complex**
  - **Real time**
  - **Packaged as a C program**
- Developed a four-class SNN classifier

ECG Compression:

- Developed the hybrid PWM-quantization ECG compression algorithm
  - **Low complex**
  - **Real time**
  - **Nearly perfect reconstruction**
  - **Packaged as a C program**

On-going: Publishing the Findings

Hybrid PWM-Quantization ECG Compression: Balancing Simplicity, Compression Ratios, and Signal Fidelity

Sashini Liyanage*
Department of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka
e17190@eng.pdn.ac.lk

Isuri Devindi*
Department of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka
e17058@eng.pdn.ac.lk

Titus Jayarathna
MARCS Institute for Brain, Behaviour and Development
Western Sydney University
Westead NSW, Australia
t.jayarathna@westernsydney.edu.au

Roshan Ragel
Department of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka
roshanr@eng.pdn.ac.lk

I. Introduction

Arrhythmias or irregular heart rhythms are a significant component of cardiovascular diseases (CVD), a leading cause of global mortality. The electrocardiogram (ECG) is a vital tool widely used among healthcare professionals to diagnose and monitor these abnormalities. While the manual inspection of ECG signals in an acute condition is a tedious task prone to human errors, long-term monitoring of the electrical activity of the heart for early detection of transient or infrequent arrhythmias is also challenging without computer-aided diagnosis methods.

Over time, many sophisticated and highly accurate computer-aided arrhythmia diagnosis methods have been in-

Even though lossless ECG compression algorithms tend to preserve signal quality well, due to the high complexity of these algorithms, they are unsuitable for real-time applications. On the other hand, while lossy compression techniques can achieve high compression ratios, they are less reliable when compared to lossless techniques. Moreover, the performance of most of the ECG compression methods is widely evaluated only on signal quality measurement metrics such as the percentage root-mean-square difference (PRD) and compression ratio (CR). However, since the goal of compressing and transmitting ECG signals is to reconstruct them for further analysis, especially for arrhythmia detection, it is crucial to assess the performance in terms of QRS detection and heart-rate variability measurements.

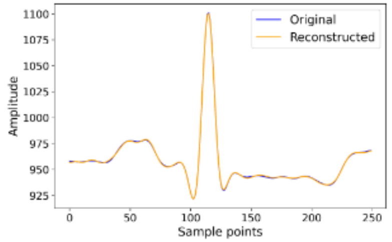


Fig. 2: The difference between the original and signal reconstructed from the 6-bit compressed signal

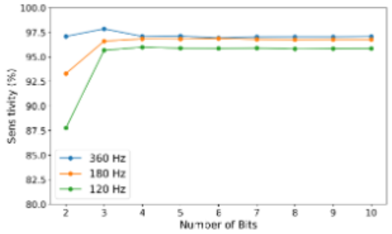


Fig. 4: QRS peak detection sensitivity performance for Pan-Tompkins Algorithm for different step sizes

D. Storing the compressed signal

The compressed signals are stored as integers in separate text files. The entire folder of 48 recordings is then compressed using the ZIP file format which is a popular archive file format for lossless compression. The compressed folder size was further reduced, by storing multiple samples in a single 64-bit word. For instance, the compressed signals in 4-bit resolution were stored by storing 16 compressed sample points in a single

Low Complexity Algorithm for Real-time QRS Peak Detection

Sashini Liyanage*
Department of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka
e17190@eng.pdn.ac.lk

Isuri Devindi*
Department of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka
e17058@eng.pdn.ac.lk

Titus Jayarathna
MARCS Institute for Brain, Behaviour and Development
Western Sydney University
Westead NSW, Australia
t.jayarathna@westernsydney.edu.au

Roshan Ragel
Department of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka
roshanr@eng.pdn.ac.lk

Furthermore, with the introduction of wearable ECG devices, the possibility of real-time heart disease detection is now made available. For instance, such an edge device can be configured to immediately detect and store a suspected abnormal beat of the ECG for subsequent transmission to a central station for in-depth analysis, either automatically by a machine learning algorithm or manually by a physician. As the QRS complex is a prominent feature used for beat segmentation of an ECG, accurate detection of QRS complexes is essential for such a wearable device.

False QRS detection may result in incorrect segmentations, unnecessary data transmission, and ultimately inaccurate detection of arrhythmia. Detecting QRS accurately is inherently difficult due to physiological variations in the complexes

aiming for swift and precise QRS detection with minimal memory and time consumption is even more challenging. In this paper, we propose a low-complexity algorithm for real-time detection of the QRS peaks in an ECG signal that can be implemented in severely resource-constrained environments.

The two main stages that can be identified in all QRS detection algorithms introduced over the years are the denoising stage using a combination of filters (e.g. highpass and lowpass, or bandpass) to detect the prominent features of the QRS complex such as amplitude and width and the decision making stage to detect the R-peaks using a threshold. The proposed algorithm optimizes this workflow by eliminating the redundant memory utilization and complexity introduced by filters in the feature extraction stage. It employs a simple peak-to-peak envelope detection technique to derive the signal's height feature, ensuring immunity to disturbances like baseline wander and minimum latency. Furthermore, it employs a rule-based approach to eliminate false detections considering the dynamic variation of the heart rate of the ECG signal. This work also utilizes a dynamic threshold sensitive to the height of the R-peaks with high robustness to noise. The proposed work was evaluated for noise robustness, low resource consumption (memory and computational power), and low latency for real-time detection on 3 public databases.

As this optimized solution demonstrates improved performances compared to other existing methods, it can be utilized to develop affordable wearable devices for real-time cardiac health assessments for a broader population.

II. Related Works

The most difficult challenge for QRS detectors is accurately spotting heartbeats when there are noise and unwanted signals. This detection process usually involves two main steps: first,

Write **three research papers**

R-peak detection, classification, compression

Deploy an open-source arrhythmia detection software containing functions for R-peak detection, signal compression, heart-beat classification

Intended publisher: IEEE Journal of Biomedical and Health Informatics

Expected Outcomes & Deliverables

Pre-packaged arrhythmia detection software

Low-complexity algorithm to detect different arrhythmia conditions from ECG signals

Compatible with resource-limited microcontrollers

Thank You !

Q & A

Expected Outcomes & Deliverables

R-peak Detection:

- Developed the leaky-boundary based R-peak detection algorithm
  - **Low complex**
  - **Real time**
  - **Packaged as a C program**

Beat Classification:

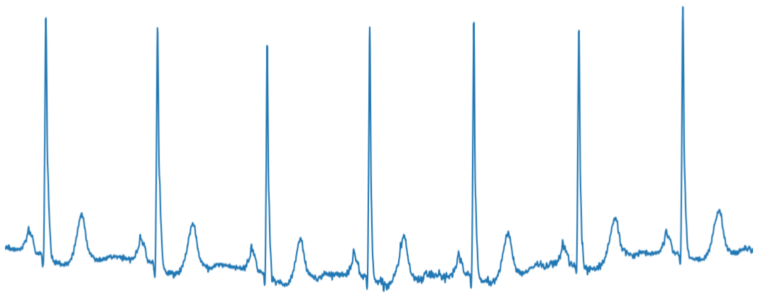
- Developed an on-edge device rule-based binary classifier
  - **Low complex**
  - **Real time**
  - **Packaged as a C program**
- Developed a four-class SNN classifier

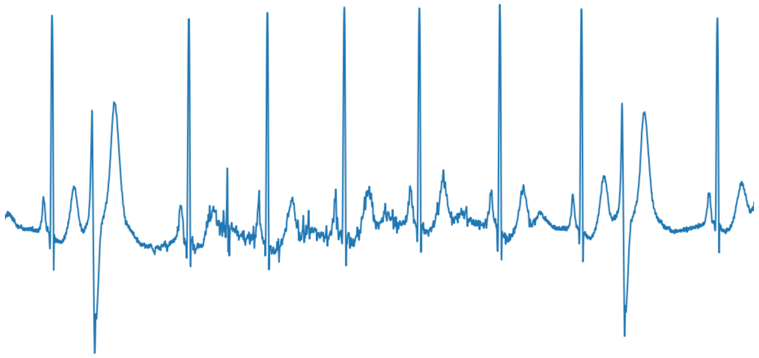
ECG Compression:

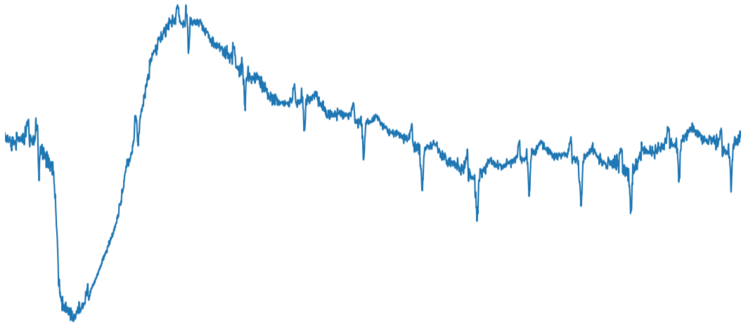
- Developed the hybrid PWM-quantization ECG compression algorithm
  - **Low complex**
  - **Real time**
  - **Nearly perfect reconstruction**
  - **Packaged as a C program**

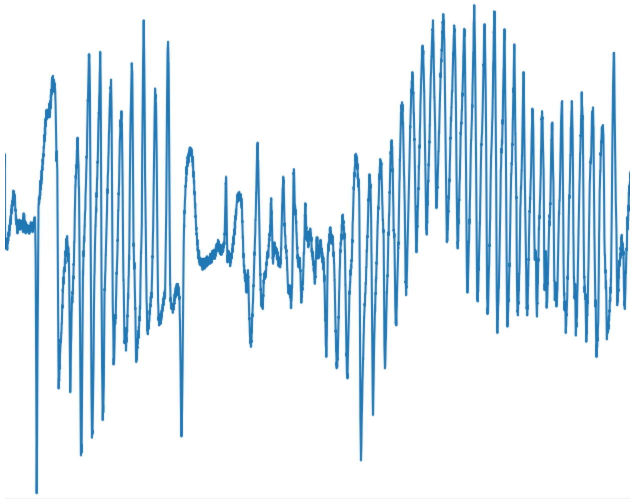
ECG Data

MIT-BIH Arrhythmia Database
48 half-hour labeled ECG signals









Experiment: Leaky Boundary R-Peak Detection

Operation	Number of Integer Operations
Division	2
Multiplication	5
Subtraction	9
Addition	10
Comparison	3
Total	29

Number of Integer Operations Per Sample Point