JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES
www.journalimcms.org

ISSN (Online) : 2454 -7190 Vol.-21, No.-11, November (2026) pp 166-186 ISSN (Print) 0973-8975

COMPARATIVE STATISTICAL ANALYSIS AND PREPROCESSING USING MIT-BIH ARRHYTHMIA DATABASE AND AD8232

Tribedi Sarma[1] , Rajkishur Mudoi[2]

[1,2] North-Eastern Hill University, Shillong, Meghalaya, 793022, India.

Email: [1]tribedi.81@gmail.com, [2]rajkishur@gmail.com

Corresponding Author: **Rajkishur Mudoi**

https://doi.org/10.26782/jmcms.2026.11.00008

(Received: August 31, 2026; Revised: October 30, 2026; Accepted: November 12, 2026)

Abstract

Electrocardiogram (ECG) signals are extensively used for the diagnosis of patients and the detection of cardiovascular abnormalities. However, acquired ECG signals from patients are often influenced by noise, baseline drift, DC offset, and amplitude fluctuations, which may affect the processing of the signal for further analysis. This study presents a hardware-based ECG acquisition and analysis framework that includes an AD8232 ECG sensor and an Arduino Nano. ECG signals are acquired from six participants and transferred to a Python-based software environment for further analysis. The acquired ECG signals collected from both hardware and the MIT-BIH Arrhythmia Database are standardized and normalized, followed by DC offset removal and Butterworth band-pass filtering with a frequency range of 0.5–40 Hz. These preprocessing steps will reduce baseline wander and high-frequency noise while retaining the major components of the ECG waveform. The ECG waveforms from both data sources are visualized at each processing stage, and statistical analysis is performed on the processed signals. The proposed framework provides an integrated approach for hardware-based ECG acquisition, signal preprocessing, visualization, and statistical analysis using both real-world and benchmark ECG data.

Keywords: ECG, MIT-BIH Arrhythmia Database, AD8232, Arduino Nano, ECG Signal Processing, DC Offset Removal, Butterworth Filter.

I. Introduction

Cardiovascular diseases remain one of the leading causes of mortality worldwide. This necessitates continuous monitoring of cardiac electrical activity for timely detection and intervention of cardiac abnormalities [VI]. The electrocardiogram (ECG) is an important non-invasive tool for evaluating cardiac rhythm and detecting various abnormalities such as arrhythmias, ischemia, and conduction disorders. The ECG waveform consists of several characteristic components, namely the P wave, QRS complex, and T wave, that provide important

information about cardiac electrical activity [XIX] [XIII]. Although traditional ECG acquisition systems offer high signal quality, they are limited in accessibility for continuous or remote monitoring. The availability of low-cost electronic components and computational techniques has made it easier to develop portable ECG acquisition and automated analysis systems for research and remote-monitoring applications [XIV] [XX]. Several studies have explored various Arduino-based ECG acquisition systems for real-time ECG signal acquisition and automated cardiac abnormality detection [XX] [X]. However, such low-cost hardware is affected by various noise sources such as baseline wander from respiration and electrode motion, powerline interference, and electromyographic (muscle) noise, compared to clinical-grade acquisition systems. These artefacts distort waveform features, such as the P wave, QRS complex, and T wave, which reduces the accuracy of heart rate estimation and arrhythmia classification.

The MIT-BIH Arrhythmia Database is widely used as a standard clinical database that contains half-hour, two-channel ECG recordings, sampled at 360 samples per second per channel. It is well-annotated, high-quality signals widely used as reference data for signal processing and machine learning algorithms [XV] [XVIII]. Several existing studies have integrated low-cost AD8232/Arduino-based acquisition with MIT-BIH-trained models, preprocessing the hardware-acquired signal and then evaluating it through classifier performance [XIX] [V]. This approach, however, is still unclear about how similar a low-cost ECG acquisition system is to standard clinical ECG data after applying the same preprocessing steps to both datasets. To address this gap, the proposed study applies a standardized preprocessing pipeline to both MIT-BIH reference signals and ECG data acquired from a custom AD8232 and Arduino Nano-based hardware system. Both signal sources are standardized, corrected for DC offset, and filtered using a Butterworth band-pass filter with a frequency range 0.5–40 Hz. This selected filter retains the important ECG waveform components while reducing baseline wander and high-frequency noise. The preprocessed signals are then compared using statistical analysis to evaluate the fidelity of the hardware-a cquired data relative to the clinical reference.

The main contributions of this study are as follows.

- Downloaded the MIT-BIH dataset.
- Recorded ECG signal from hardware.
- Converted .dat file into .csv from MIT-BIH (all .dat to .csv).
- Standardized the MIT-BIH dataset.
- Standardized hardware-collected dataset.
- Preprocessed MIT-BIH dataset (From Standardized dataset) (DC offset removal).
- Preprocessed hardware-collected dataset (from standardized dataset) (DC offset removal).
- Implemented a Butterworth band-pass filter (0.5–40 Hz).
- Plotted ECG graphs.
- Compared statistical results.

The remainder of this paper is organized as follows: Section 2 reviews related work on low-cost ECG acquisition and preprocessing techniques; Section 3 details the materials and methods, including hardware setup and the preprocessing pipeline; Section 4 presents the ECG graphs and comparative statistical results; and Section 5 concludes with directions for future work.

II. Motivation

ECG signals collected using low-cost hardware systems may be affected by DC offset, baseline variations, noise, and amplitude fluctuations. These unwanted components may reduce the quality of the signal and make it difficult for further analysis. Although previous studies have extensively used MIT-BIH and other standard ECG databases for signal analysis, very few studies have covered a complete workflow from raw ECG acquisition to preprocessing, filtering, visualization, and statistical analysis of real hardware-acquired ECG signals. This research gap motivated the present work, in which ECG signals are acquired from six subjects using an AD8232 sensor and Arduino Nano and processed through DC offset removal, Z-score normalization, and Butterworth band-pass filtering. The same processing steps are also applied to the MIT-BIH ECG Arrhythmia database, and the resulting signals obtained from both sources are visualized and statistically analyzed. Thus, the proposed work provides a systematic and reproducible ECG acquisition and preprocessing framework that facilitates further analysis of the ECG signal.

III. Literature Survey

Many researchers have focused on the development of low-cost signal acquisition hardware with automated signal processing and machine learning techniques to enable portable and real-time cardiac monitoring. A low-cost analog front-end device, such as the AD8232, integrated with microcontrollers, such as Arduino, provides a practical foundation to develop compact ECG devices. In addition, publicly available datasets such as the MIT-BIH Arrhythmia Database are used to provide benchmark data for algorithm development and evaluation.

Freudenberg et al. have reviewed a systematic literature survey of various real-time ECG preprocessing methods. A Butterworth band-pass filter with a range of 0.1–100 Hz has been chosen as the standard approach for future analysis. Various preprocessing steps are analyzed in this paper [IX].

Abdrasulova et al. systematically compared 4th-order Butterworth and Elliptic IIR bandpass filters at multiple sampling frequencies, such as 1000, 500, and 250 Hz. The comparative study finds that Butterworth filtering at 500 Hz has achieved strong similarities with ECG-derived reference values, while Elliptic filtering showed greater distortion. Hence, the Butterworth filter is considered a better filter than the Elliptic IIR filter to process the ECG signal [I].

Gupta et al. developed Heart Beat AI, a multi-label ECG arrhythmia detection framework that applies a second-order Butterworth bandpass filter spanning 0.5–45 Hz as its preprocessing stage, explicitly noting that the 0.5 Hz lower cutoff attenuates

baseline drift while the upper cutoff preserves QRS and P-wave morphology — a passband nearly identical to your own 0.5–40 Hz choice [VIII].

He et al. applied a fourth-order Butterworth bandpass filter at 0.67–40 Hz following power-line notch filtering in a wearable single-lead ECG framework for structural heart disease detection, correcting residual baseline wander via median-filter subtraction rather than DC offset removal — a useful alternative baseline-correction method to reference in your discussion [XI].

Dündar Ö et al. presented a portable ECG device based on the AD8232 sensor and an embedded processing platform for continuous cardiac monitoring. Their study used the MIT-BIH Arrhythmia Database for model development and additionally collected ECG data from 20 individuals, including healthy and arrhythmic subjects, using the proposed hardware system. The ECG recordings are segmented into 60-second intervals, preprocessed, normalized, and analysed. The results obtained from this pipeline are used to extract time-domain and statistical features before machine-learning classification. [III].

IV. Materials and Methods

Hardware Requirements

Table 1: Hardware Component Details of the ECG Acquisition System.

Hardware Component	Chose Device	Specifications	Estimated Price (INR)
ECG Signal Conditioning Unit	AD8232	Lead ECG analog front-end; power supply: 2.0 to 3.5V; typical current consumption: 170uA; common mode rejection ratio (CMRR): ~ 80dB; temperature range: -40C to +85C [XVI] [XII].	850
Micro-controller/Data Acquisition unit	Arduino Nano (ATmega328)	8-bit AVR RISC based microcontroller, Vop = 1.8 - 5V, Max Clock= 20 MHz, Flash memory = 32KB, SRAM = 2KB, EEPROM =1KB [X] [II].	450

Software Requirements

Python is selected as the primary software platform to simplify and standardize the different computational tasks involved in ECG acquisition, storage, preprocessing, visualization, comparison, and classification of ECG signals. Instead of developing every mathematical, signal-processing, database, and machine-learning operation from the beginning, established libraries provide tested functions that can be integrated into the proposed system. The following Python libraries are used in the ECG processing pipeline.

Table 2: Python Libraries Utilized in the Proposed ECG Analysis System.

Python library	Why it is used in this study	Application
NumPy [IX]	It allows complex mathematical operations, manipulation of numerical arrays, and numerical operations	Analyze and normalize ECG samples, compute mean, standard deviation, root mean square (RMS) values, and other intricate mathematical operations.
Pandas	It offers functions and structures for easier manipulation of data	Used it to store, clean, manipulate, order, and convert our readings of ECG signals into CSV files.
Matplotlib	Display ECG waveform	Normalized, filtered, and different comparing ECG signal plots.
WFDB [XVIII]	Designed specifically to read physiological databases	Reading MIT-BIH ECG records, retrieve signal information, labels, and annotations
SciPy [XVII]	Provides general-purpose scientific computing and technical computing.	Remove DC offset, use a Butterworth filter, design a bandpass filter, and perform signal analysis

Workflow of the System

The proposed study consists of two ECG data sources: (i) ECG signals acquired from a low-cost hardware system and (ii) ECG signals obtained from the MIT-BIH Arrhythmia Database. A common preprocessing methodology is applied to both sources before comparative analysis and classification. Fig. 1 illustrates the entire workflow of the proposed system that consists of two parallel data processing paths. One path corresponds to the ECG signal acquired from hardware (AD8232 interfaced with Arduino Nano), and the other corresponds to the ECG signal taken from the MIT-BIH database. The hardware branch (AD8232 + Arduino Nano) involves standardization, DC offset removal, and Butterworth band-pass filtering (0.5–40 Hz). An intermediate verification step has been done to compare the signal obtained from DC-offset removal with the original raw waveform to verify the effectiveness of baseline correction. The MIT-BIH branch follows a corresponding workflow in which initially the raw dataset is converted from .dat format to .csv, followed by standardization, DC offset removal, and the same band-pass filter is applied. This common processing method ensures both signal sources are processed under comparable conditions.

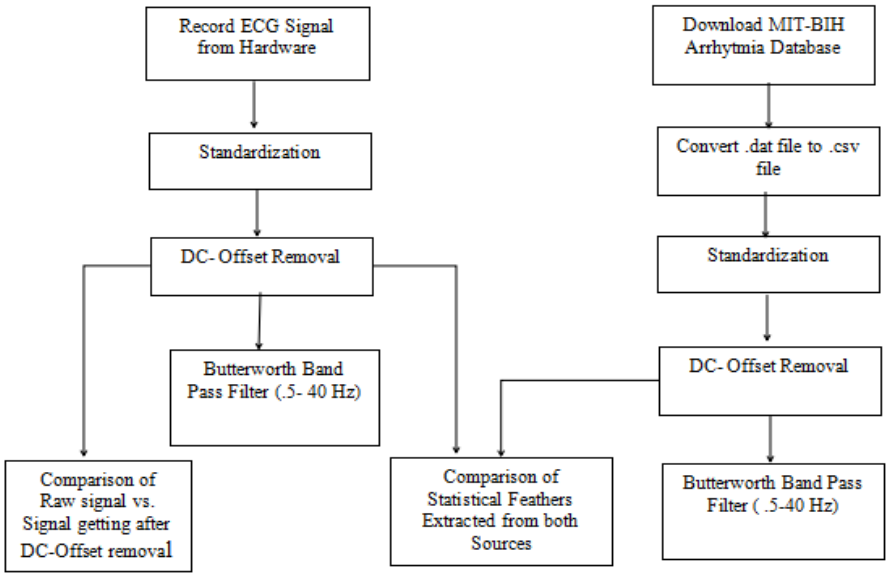


Fig. 1. Dual-branch preprocessing pipeline applied to hardware-acquired and MIT-BIH reference ECG signals, presenting the entire workflow of the system.

Hardware Block Description

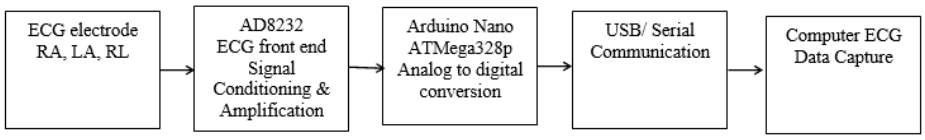


Fig. 2. Block diagram of the proposed hardware-based ECG signal acquisition system.

Fig. 2 shows a block description of the system. The system consists of four primary stages, including ECG electrodes, an ECG front-end based on the AD8232 integrated chip, an Arduino Nano module, and a computer system for serial-based receiving of ECG data. The ECG electrode pair receives the electrical impulses created by the heart and presents the bio-current signal to the ECG front-end AD8232. The AD8232 ECG signal conditioner works as an ECG signal interface by getting the tiny bio-electrical impulses from electrodes and processing them for amplification into the desired analog ECG signal output. Arduino Nano gets the processed analog ECG signal from the conditioning unit, then converts the analog to digital via its ADC block at a sampling frequency of 360 Hz. The resulting digitized values will be sent to a computer via serial data communication based on the USB port. This data will be recorded using some ECG recording modules under a programmed data capture language (Python software) and stored in a database for further signal preprocessing, visualization, comparison with MIT-BIH ECG recordings, and classification. Thus, the hardware architecture provides the interface between the physiological ECG signal and the proposed computer-based ECG analysis framework.

Real Dataset

The raw data collected from the hardware includes both males and females having age group of 25-75 years with no known cardiac history. Ethical approval has been applied for and approved by one of the leading healthcare institutes, GNRC, Dispur, Guwahati, Assam, so that a greater number of clinical data can be collected with known cardiac conditions, supervised by medical experts.
As the earlier data have a limited number of subjects and are collected under a non-medical environment, it is not very accurate and difficult to group into similar rhythms unless there is no known cardiac history. After collecting the data from real patients with known cardiac history, it will be possible to categorize and compare the data of similar rhythms.

MIT-BIH Arrhythmia Database

The MIT-BIH Arrhythmia Database is considered a reference ECG dataset. The database consists of 30 minutes and 6 seconds of two-channel ambulatory ECG signals recorded from 47 subjects. The recordings have been digitized at 360 samples per second per channel with 11 bits of resolution, with a ±10 mV voltage range, and have reference annotations for individual beats. The original MIT-BIH dataset is available in the form of signal (.dat), header (.hea), and annotation (.atr) files. The required ECG recordings are acquired through Python programming. Selected ECG channels of the records are extracted and converted to CSV format.

Conversion of MIT-BIH Records to CSV

The MIT-BIH signal data are loaded via the use of a suitable Python library that is compatible with ECG/PhysioNet. The ECG signals are separated from the chosen channel and arranged in a tabulated form. Fig. 3 represents an MIT-BIH ECG record that has been converted from .dat to .csv format, with the following four columns: Sample (Row Index), Time_sec (timestamp that increases by ~0.002777 s, which confirms the typical sampling rate of 360 Hz used in MIT-BIH), MLII (Modified Limb Lead II), and V1 (chest lead) measured in mV.

Sample	Time_sec	MLII	V1
0	0	-0.345	-0.16
1	0.00277777	-0.345	-0.16
2	0.00555555	-0.345	-0.16
3	0.00833333	-0.345	-0.16
4	0.01111111	-0.345	-0.16
5	0.01388888	-0.345	-0.16
6	0.01666666	-0.345	-0.16
7	0.01944444	-0.345	-0.16
8	0.02222222	-0.33	-0.15
9	0.025	-0.32	-0.145
10	0.02777777	-0.32	-0.17
11	0.03055555	-0.325	-0.18
12	0.03333333	-0.33	-0.17
13	0.03611111	-0.325	-0.15
14	0.03888888	-0.315	-0.16
15	0.04166666	-0.305	-0.17
16	0.04444444	-0.3	-0.185
17	0.04722222	-0.315	-0.195
18	0.05	-0.32	-0.185
19	0.05277777	-0.315	-0.185
20	0.05555555	-0.3	-0.175

Fig. 3. Sample of the MIT-BIH arrhythmia database record after conversion from .dat to .csv format

Dataset standardization

As ECG data was obtained from two different systems with different sampling conditions, i.e., one ECG dataset was obtained from the MIT-BIH Arrhythmia Database and the other was acquired using the developed AD8232-based hardware system. Hence, it is necessary to standardize both datasets into a tabular format so that they can be processed using the same analysis and machine-learning pipeline. Both tables contain three columns, namely sample, Time_ms, and ECG, as shown in Fig. 4. Both data sources have the same sampling frequency, sequential sample numbering, and time representation, making them easier to be compatible for analysis.

Sample	Time_ms	ECG
0	0	279
1	2.778	277
2	5.556	276
3	8.333	275
4	11.111	274
5	13.889	274
6	16.667	273
7	19.444	274
8	22.222	275
9	25	276
10	27.778	277
11	30.556	277
12	33.333	277
13	36.111	277
14	38.889	279
15	41.667	281
16	44.444	282
17	47.222	282
18	50	283
19	52.778	285
20	55.556	286

(a)

Sample	Time_ms	ECG
0	0	-0.145
1	2.77777777	-0.145
2	5.55555555	-0.145
3	8.33333333	-0.145
4	11.1111111	-0.145
5	13.8888888	-0.145
6	16.6666666	-0.145
7	19.4444444	-0.145
8	22.2222222	-0.12
9	25	-0.135
10	27.7777777	-0.145
11	30.5555555	-0.15
12	33.3333333	-0.16
13	36.1111111	-0.155
14	38.8888888	-0.16
15	41.6666666	-0.175
16	44.4444444	-0.18
17	47.2222222	-0.185
18	50	-0.17
19	52.7777777	-0.155
20	55.5555555	-0.175

(b)

Fig. 4. Standardized ECG data from (a) MIT-BIH dataset and (b) Arduino–AD8232-based data

ECG Signal Preprocessing

In the preprocessing phase, as shown in Fig. 5, both MIT-BIH and Arduino ECG signals follow the identical basic Preprocessing steps that will transform the ECG signals to a state where they can be used for analysis and machine learning applications. Both signals are subjected to the removal of the DC offset (by subtracting its mean value). After removal of the offset, Z-score normalization has been applied to both signals using the corresponding values of the mean and standard deviation. ECGs are pre-processed in batches automatically in two output folders following similar preprocessing routines.

Sample	Time_ms	ECG
0	0	0.18192237537356293
1	2.77777777	0.1786376880418006
2	5.55555555	0.1780337454586379
3	8.33333333	0.1823660025648221
4	11.1111111	0.19285638722390044
5	13.8888888	0.20889655236524154
6	16.6666666	0.22750231371484575
7	19.4444444	0.2436195197989549
8	22.2222222	0.2515840086505801
9	25	0.24727450883876242
10	27.7777777	0.2298233989310894
11	30.5555555	0.20189348095411777
12	33.3333333	0.1684859094333835
13	36.1111111	0.13512398100109352
14	38.8888888	0.10631346629596844
15	41.6666666	0.0846402027234647
16	44.4444444	0.07050191225849764
17	47.2222222	0.06258119571759141
18	50	0.05921678727626931
19	52.7777777	0.06021387567234516
20	55.5555555	0.06770371275520111

(a)

Sample	Time_ms	ECG
0	0	-0.171633
1	2.778	-0.175969
2	5.556	-0.177922
3	8.333	-0.175795
4	11.111	-0.169196
5	13.889	-0.159395
6	16.667	-0.149134
7	19.444	-0.141691
8	22.222	-0.139393
9	25	-0.142289
10	27.778	-0.147876
11	30.556	-0.152291
12	33.333	-0.152359
13	36.111	-0.147248
14	38.889	-0.138653
15	41.667	-0.129536
16	44.444	-0.122418
17	47.222	-0.118345
18	50	-0.116953
19	52.778	-0.117244
20	55.556	-0.118373

(b)

Fig. 5. ECG signal preprocessing using DC offset removal and Z-score normalization: (a) MIT-BIH ECG data and (b) Arduino–AD8232-based data.

- **DC-Offset Removal:** DC offset represents the constant or slowly varying component present in an acquired ECG signal. It can originate from the acquisition electronics, electrode-skin interface, or measurement conditions. To remove the constant DC component, the mean of the ECG signal is subtracted from each sample.

$$x_{\text{DC removed}} = x - \mu \tag{1}$$

where x is the original ECG sample, and μ is the mean (DC component) of the ECG signal.

- **Normalization:** Normalization is used to reduce differences between the amplitudes of signals for standardization of computer-supported calculations. For Z-score normalization, the following formula is used,

$$z = \frac{x-\mu}{\sigma} \tag{2}$$

the symbols mean:

- x = The actual ECG signal's sample
- μ = The mean (average) of the ECG's samples
- σ = The standard deviation of the ECG samples
- z = The standardized ECG's value

Butterworth Band-Pass Filtering

A digital Butterworth band-pass filter is implemented to suppress unwanted frequency components and preserve the principal ECG morphology. The selected pass-band is 0.5–40 Hz. The lower cutoff is selected to attenuate very-low-frequency baseline variations, while the upper cutoff is selected to reduce higher-frequency noise.

The Arduino and the MIT-BIH ECG signals both undergo preprocessing with a Butterworth bandpass filter from 0.5 Hz to 40 Hz in order to remove undesired components without significantly degrading the ECG signal features important for machine learning tasks. The Butterworth high-pass filter with a cut-off of 0.5 Hz allows the removal of low-frequency disturbances such as motion artefacts, baseline wandering during breathing, variations in the contact of the electrode to the skin, or movements of the electrode. Whereas the Butterworth low-pass filter with a cut-off at 40 Hz permits removal of high-frequency noise such as muscle contractions or electrical noise. The Butterworth filter is utilized as the pass-band has a flat frequency response that causes only minor distorting effects on the shape of the heartbeats. Applying the same processing steps to the Arduino-collected ECG signal and the MIT-BIH ECG heart signals provides a comparable signal representation for further analysis.

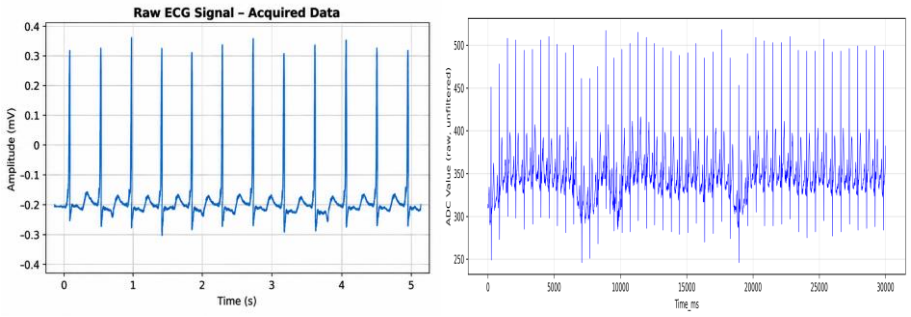
Comparative Statistical Analysis of ECG Signals

After preprocessing, the hardware-acquired and MIT-BIH ECG signals are compared using graphical and quantitative analysis based on the average values of six recordings from each data source. Both datasets use various statistical features such as mean, variance, standard deviation, RMS, power, etc., to evaluate quantitative analysis. During a comparable study between two groups, subject-level data have not been matched for data collection. It is not possible to physically collect the physiological data of the subject whose data has been considered for the MIT-BIH dataset. The main objective of this work is to normalize the amplitude of the parameters instead of comparing individual data. Hence, the average of six subjects has been taken. The primary objective is to generate and normalize the processed data for the study. Moreover, statistical feature extraction will be useful in further analysis using machine learning techniques.

V. Results

The processing of ECG signals is presented in four successive stages.

- **Raw Signal**



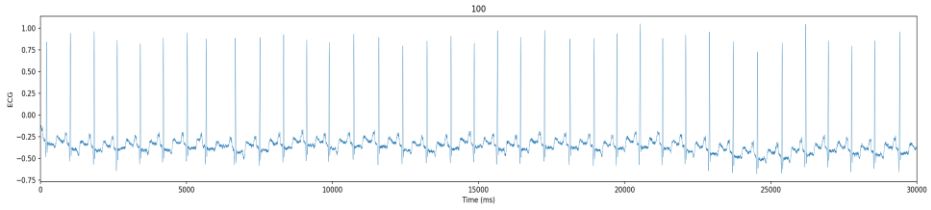
(a) (b)

Fig. 6. Raw ECG signal from (a) MIT-BIH ECG data and (b) Arduino–AD8232-based data.

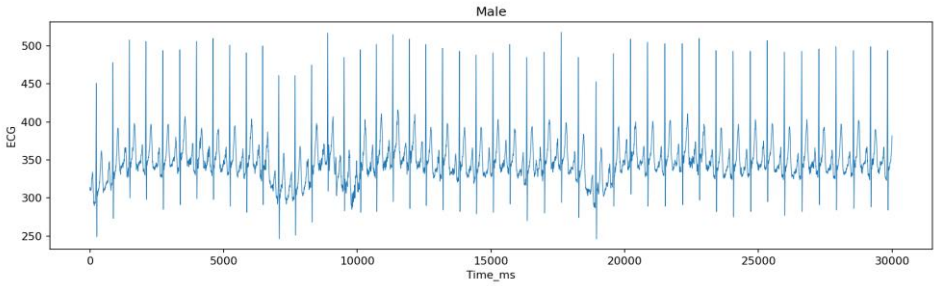
Fig. 6(a) shows a raw ECG signal recorded from the MIT-BIH Arrhythmia database, and it is plotted on the X-axis (amplitude in millivolts) against the Y-axis (Time in seconds). The repeated QRS complex peaks are clearly identifiable, with larger R-peaks being repetitive during the recorded trace; thus, the cardiac cycle is periodically present in the ECG. The baseline of the ECG is not stable, and the signal also varies and has noise at each interval between beats. The range of the amplitude is between 0.6 mV and 1.0 mV, and thus the signal has a significant DC/baseline level and amplitude variations. The raw signal can be further used to remove DC offset, normalize using Z-score, and filter using a Butterworth band-pass filter. The signal is basically recorded for 30 seconds. Once the entire signal is plotted for 30 seconds, the entire waveform will not be visible clearly. So, only 12- second duration data is plotted so that spikes are clearly visible.

Fig. 6(b) shows the original ECG signal recorded by the Arduino/AD8232 hardware system before any processing. It is plotted on the X-axis (time in ms) and the Y-axis (amplitude). The repetition of sharp peaks at certain intervals (approx. 30,000 samples) shows the R-peak each time during the cardiac cycle. There are some negative variations visible in the regions immediately after some of the major peaks, and they are mainly due to the noisy signal.

- **Standardizing Signals**



(a)



(b)

Fig. 7. Standardization of (a) MIT- BIH data and (b) Arduino–AD8232-based data.

Fig. 7(a) shows the standardized MIT-BIH ECG signal before removing the DC offset and applying Z-score normalization. The raw signal is converted from seconds to milliseconds, plotted on the x-axis, and from millivolts to ECG, plotted on the y-axis.

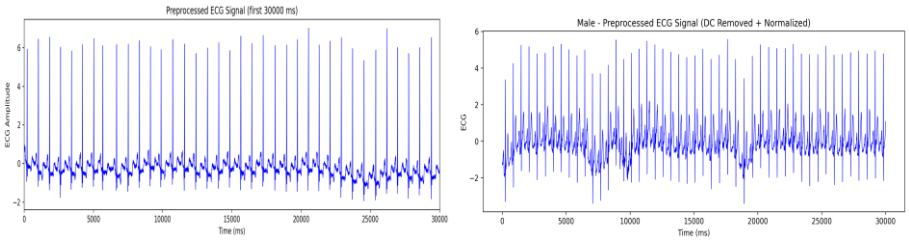
Similarly, in Fig. 7(b), the raw Arduino signal has been standardized before DC offset removal and normalization. The signal is converted from ADC to ECG, plotted on the y-axis, and the x-axis remains the same as before.

This standardization step ensured that both signals are now compatible with each other and can be processed for further steps. The standardized signals are now subjected to DC-Offset removal and normalization in the next phase.

- **Preprocessing Signal**

Fig. 8(a) shows the preprocessed MIT-BIH ECG signal after removing the DC offset and applying Z-score normalization. The signal is now centered near the zero line, and the values have been normalized on a standard scale, mostly between +0.3 and -0.3, where prominent R-peaks can still be distinguished, reaching near +1.0, thus giving normalized variation in the amplitude along with some fluctuation and noise. Hence, the Butterworth band-pass filter (0.5-40 Hz) is later applied in the following step.

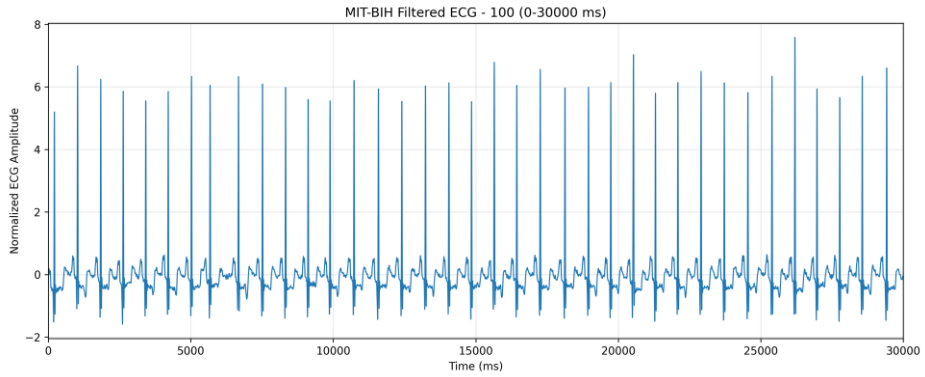
Fig. 8(b) is the preprocessed ECG signal after DC-offset removal and normalization. The waveform has been centered near the zero level and standardized to obtain a consistent representation of the ECG across the recorded period. The recurring sharp peaks are indicative of the R-peaks of successive heartbeats. The repeated sharp peaks correspond to the R-peaks of successive heartbeats, while the smaller fluctuations between the peaks represent the other components of the ECG waveform. Although the signal's amplitude variation has been normalized, noise and other fluctuations can be seen. Hence, a Butterworth band-pass filter (0.5-40 Hz) is applied in the next stage.



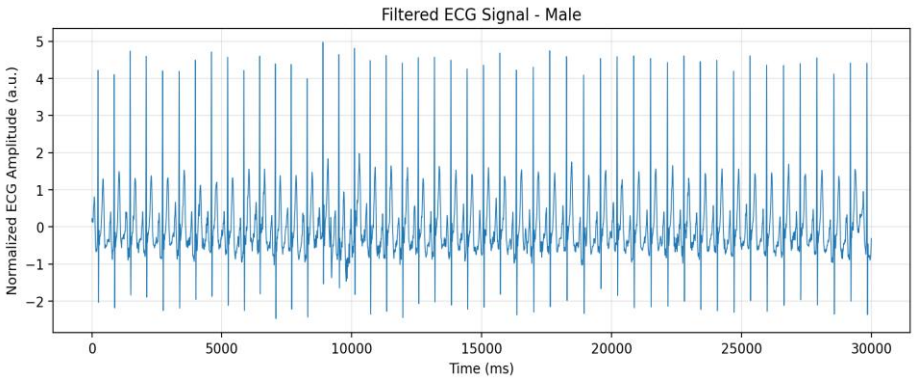
(a) (b)

Fig. 8. Preprocessed ECG signals from (a) the MIT-BIH dataset and (b) Arduino-based data.

- **Butterworth Filter Graphs**



(a)



(b)

Fig. 9. Butterworth band-pass filtered ECG signals from (a) the MIT-BIH dataset and (b) Arduino-based data.

Fig. 9(a) shows the MIT-BIH ECG signal after applying a Butterworth band-pass filter with a frequency range of 0.5–40 Hz. The filtered waveform is centered approximately around zero, while the prominent R-peaks remain clearly visible, indicating that the main cardiac features of the ECG have been preserved. The

baseline fluctuations and high-frequency variations present in the earlier processed signal have been reduced, resulting in a smoother waveform between successive heartbeats. This filtering stage effectively removes unwanted frequency components while retaining the important morphological characteristics of both ECG signals.

Fig. 9(b) presents an obvious and smooth filtered waveform for the entire recorded session, showing recurrent sharp peak patterns as the QRS complexes, as well as other waveform components as the cardiac cycle information. The Butterworth band-pass filter suppresses noise and DC drift in the signals and highlights a certain bandwidth region, which is useful to be applied for beat detection, heart-rate calculation, RR-interval analysis, and statistical feature extraction. The presence of repeated and distinguishable cardiac cycles across the recording indicates that the acquired signal retains useful ECG morphological information after preprocessing.

Statistical Analysis

Table 3: Comparative statistical analysis of six ECG recordings from the MIT-BIH dataset and six Arduino-acquired ECG recordings.

Parameter	MIT-BIH (avg of 6)	Arduino (avg of 6)	Difference
Minimum	-6.733	-2.772	3.961
Maximum	9.182	6.355	2.827
Mean	0	0	0
Median	-0.14	-0.145	0.005
Range	15.915	9.128	6.787
Variance	0.973	1	0.027
Standard Deviation	0.987	1	0.013
RMS	0.987	1	0.013
Peak	9.19	6.355	2.835
Peak-to-Peak	15.915	9.128	6.787
Energy	566263.178	10800	555463.178
Power	0.973	1	0.027
Mode	-6.733	-0.287	6.446
Skewness	2.686	2.824	0.138
Kurtosis	13.785	12.911	0.874
Interquartile Range	0.617	0.742	0.125
Detected Beats	2132.833	47.333	2085.500
Heart Rate	80.117	94.667	14.550
Average RR	0.764	0.645	0.119
Minimum RR	0.313	0.375	0.062
Maximum RR	1.44	0.764	0.676
HRV (SDNN)	0.156	0.1	0.056

Table 3 summarizes the statistical characteristics obtained from the Arduino-based acquisition system with respect to the ECG signals from the MIT-BIH Arrhythmia Database for six recorded samples from each of the data sets. In this table, all of the obtained average values for mean, variance, standard deviation, RMS value, and power tend to be about standardized values of 0 and 1, respectively, reflecting the effect of normalization to preprocess the data. As stated earlier, the amplitude range of MIT-BIH signals is wider than Arduino signals, with 15.915 and 9.128, and the median values are -0.14 and -0.145. The values for skewness are higher for Arduino signals compared to MIT-BIH signals, as they are 2.824 and 2.686, respectively. The average estimated heart rates are close to one another (80.117 bpm for MIT-BIH and 94.667 bpm for Arduino). The average RR intervals are close to one another: 0.764 s for MIT-BIH and 0.645 s for Arduino. It has been found that the average HRV is lower for the Arduino-based measured ECG signal than the MIT-BIH, with values of 0.1s and 0.156s, respectively. The third column of the table quantifies gain and amplitude differences between hardware and reference recordings using calibrated peak amplitudes, RMS voltage, and beat- level amplitude distributions. Overall, the results demonstrate that the Arduino-acquired ECG signals exhibit comparable basic cardiac characteristics to the MIT-BIH signals, while some differences are observed in signal distribution and heart-rate variability.

Fig. 10 shows a comparison of the same set of standardized statistical parameters between MIT-BIH and Arduino across five panels grouped by scale. Central tendency measures (Minimum, Maximum, Median, Mode) show MIT-BIH is consistently more spread out (Minimum is almost -7, with Arduino -2.8, and max is up to 9 with Arduino 6.4), although the mean and median are very close to zero in both, which makes sense from the fact that Z-score normalization is done in both datasets. Spread metrics (Range, Peak, Peak-to-Peak) show the same pattern, with MIT-BIH roughly larger than Arduino, while Variance, Standard Deviation, and RMS stay essentially identical (~1.0) for both, as expected from normalization.

Waveform shape metrics (Skewness, Kurtosis) are close between the two sources, with Arduino only slightly higher on Skewness and MIT-BIH slightly higher on Kurtosis, suggesting broadly similar signal shape despite the amplitude differences.

Heart Rate is noticeably higher on the Arduino (~95 BPM) than on the MIT-BIH (~80 BPM). RR interval/HRV metrics demonstrate a broader Maximum and higher HRV (SDNN) in MIT-BIH data.

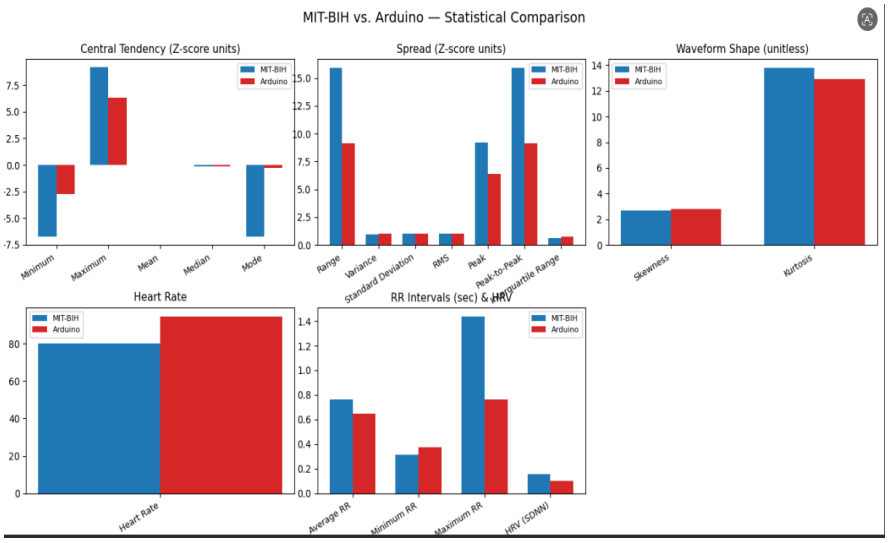
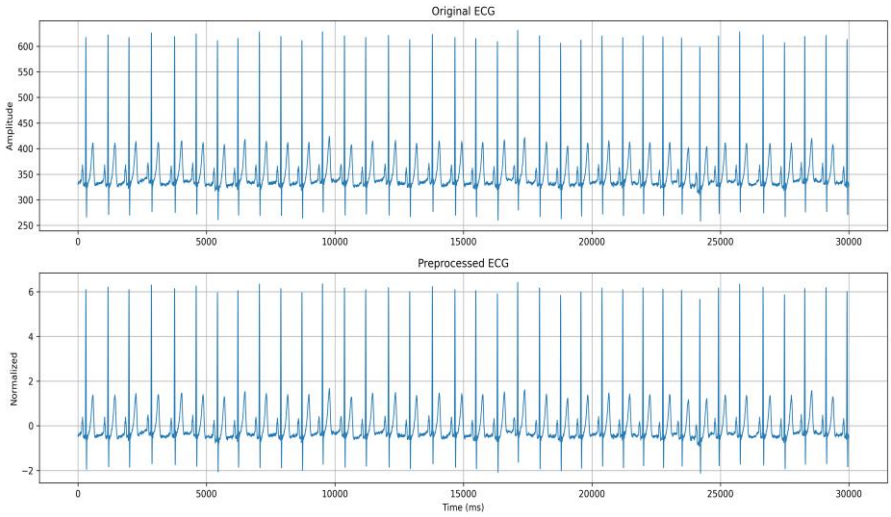
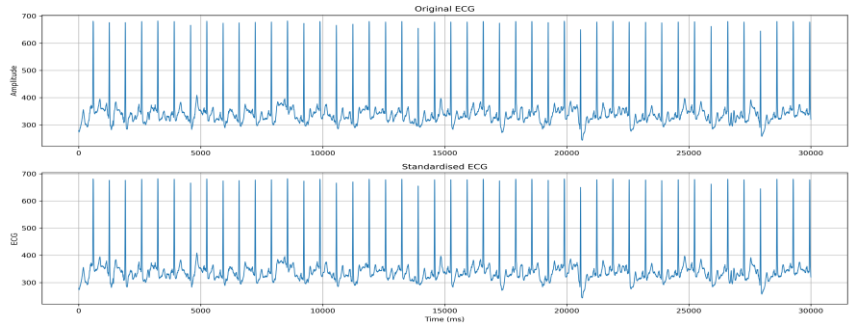


Fig. 10. shows a comparison of standardized statistical parameters (Z-score normalized) between MIT-BIH reference data and Arduino/AD8232 hardware-collected data.

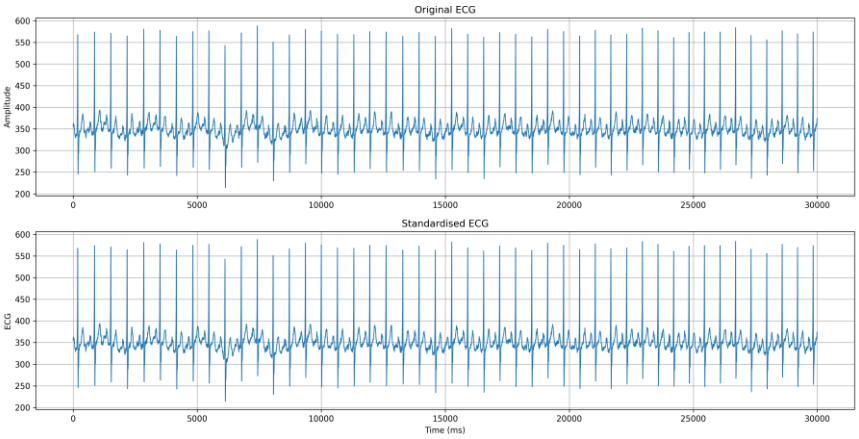
Comparative Signal Analysis



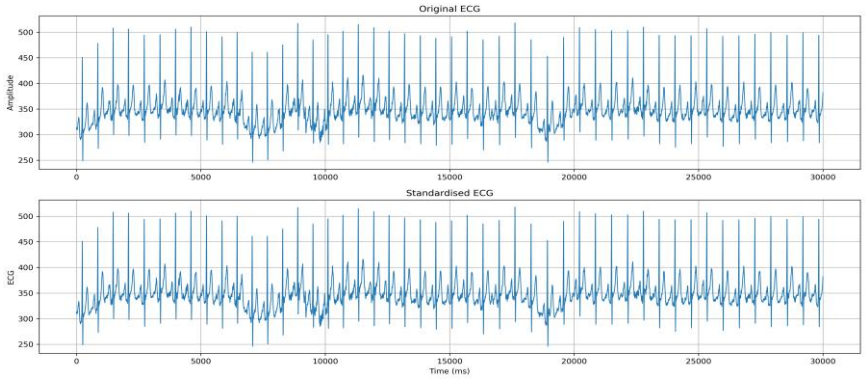
(a)


Original ECG
Amplitude
Standardised ECG
ECG
Time (ms)

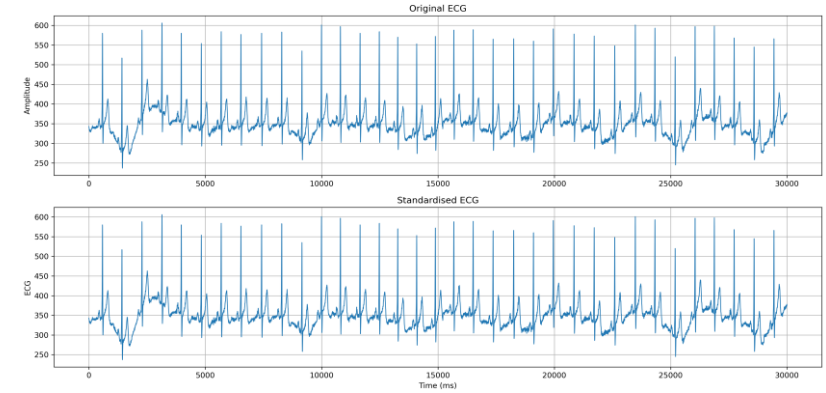
(b)


Original ECG
Amplitude
Standardised ECG
ECG
Time (ms)

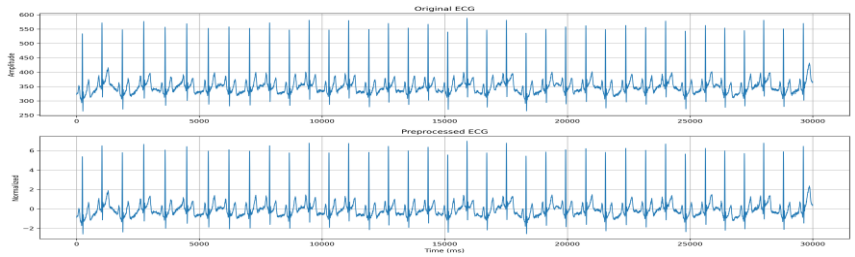
(c)


Original ECG
Amplitude
Standardised ECG
ECG
Time (ms)

(d)



(e)



(f)

Fig. 11. Comparison of raw vs preprocessed signal acquired from six subjects.

Fig. 11 shows a comparative study between original ECG data and preprocessed ECG data acquired from six subjects. In Fig. 11 (a), the top panel shows raw ADC counts, baseline wandering around 330–350, and QRS peaks up to ~630, whereas the bottom panel shows a baseline flat at 0 and QRS peaks around 6 that have the same physiological signal on completely different scales. A similar pattern is also shown in the other five graphs (Fig. 11 (b), (c), (d), (e) and (f)). Though preprocessing changes the scale of the signal, the shapes are the same. Every QRS complex in the top panel has a corresponding QRS complex in the same relative position in the bottom panel, with the same relative height ratios between beats.

VI. Conclusion

This paper systematically demonstrates an integrated method to acquire ECG signals, preprocess, statistically analyze, and visualize using both ECG signals obtained from hardware and the MIT-BIH Arrhythmia Database. ECG signals are acquired successfully from six subjects using the ECG sensor AD8232 connected to an Arduino Nano and transferred to a Python environment, which preprocesses the signals. Signals from both data sources are passed through a sequence in the entire pipeline of the system. Graphs obtained from different preprocessing tasks show that ECG signals are becoming more standard and cleaner stage by stage, with unchanged dominant characteristics of waveforms. Statistical analysis of processed ECG signals is also performed, which obtains statistical features of signals. The results showed

that the presented framework is possible to obtain and preprocess ECG signals from real-world hardware as well as benchmark data, which can be applied in further ECG analysis. In the further stage, this framework will be used for automatic diagnosis of cardiac abnormalities.

Conflict of Interest:

The authors declare no conflict of interest.

References

I. Abdrasulova, N., et al. "Butterworth Filtering at 500 Hz Optimizes PPG-Based Heart Rate Variability Analysis for Wearable Devices: A Comparative Study." *Sensors*, vol. 25, no. 22, 2025, article 7091, pp. 1-21. 10.3390/s25227091.

II. Abed, N. J., and E. B. A. Hussein. "Design and Implementation of Real Time Health Care Monitoring System Based on IoT." *Journal of Physics: Conference Series*, vol. 1818, no. 1, 2021, article 012044, 1-10. 10.1088/1742-6596/1818/1/012044.

III. Dündar, Ö., and S. Koçer. "Portable ECG Device for Continuous Heart Health Monitoring and Machine Learning Approaches for Arrhythmia Classification." *International Journal of 3D Printing Technologies and Digital Industry*, vol. 9, no. 3, 2025, pp. 737-51. 10.46519/ij3dptdi.1814384.

IV. Efunbote, M. I., et al. "Design and Development of Smart Patient Health Monitoring System Using ECG." *Moshood Abiola Polytechnic Journal of Women in Technical Education & Employment*, vol. 1, no. 1, 2023, pp. 1-10.

V. Falhi, H. M., and K. K. Hussain. "Design an Intelligent Real Time ECG Monitoring System Using Convolution Neural Network." *Revue d'Intelligence Artificielle*, vol. 37, no. 2, 2023, pp. 323-29. 10.18280/ria.370210.

VI. Frąk, W., et al. "Pathophysiology of Cardiovascular Diseases: New Insights into Molecular Mechanisms of Atherosclerosis, Arterial Hypertension, and Coronary Artery Disease." *Biomedicines*, vol. 10, no. 8, 2022, article 1938. 10.3390/biomedicines10081938.

VII. Freudenberg, J., et al. "Evaluation of Real-Time Preprocessing Methods in AI-Based ECG Signal Analysis." *2025 IEEE World AI IoT Congress (AIIoT)*, 2025, pp. 861-68. 10.1109/AIIoT65859.2025.11105222.

VIII. Gupta, S., et al. "HeartBeatAI: An Interpretable and Robust Deep Learning Framework for Multi-Label ECG Arrhythmia Detection." *arXiv*, 2026. 10.48550/arXiv.2605.24588.

IX. Harris, C. R., et al. "Array Programming with NumPy." *Nature*, vol. 585, no. 7825, 2020, pp. 357-62. 10.1038/s41586-020-2649-2.

X. Hasini, B. A., et al. "Real-Time ECG Monitoring System Using AD8232 Sensor and Arduino UNO for Biomedical Applications." *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT)*, 2025, pp. 113-18. 10.5220/0013892900004919.

XI. He, C., et al. "Wearable Single-Lead ECG Detects Fine-Grained Structural Heart Disease Through Echo-Report Supervision." *arXiv*, 2026. 10.48550/arXiv.2606.09332.

XII. Heaney, J., et al. "Internet of Things-Based ECG and Vitals Healthcare Monitoring System." *Micromachines*, vol. 13, no. 12, 2022, article 2153. 10.3390/mi13122153.

XIII. Mantravadi, A., et al. "CLINet: A Novel Deep Learning Network for ECG Signal Classification." *Journal of Electrocardiology*, vol. 83, 2024, pp. 41-48. 10.1016/j.jelectrocard.2024.01.004.

XIV. Panwar, A., et al. "Integrated Portable ECG Monitoring System with CNN Classification for Early Arrhythmia Detection." *Frontiers in Digital Health*, vol. 7, 2025, article 1535335, pp. 1-13. 10.3389/fdgth.2025.1535335.

XV. Pollard, T., et al. "PhysioNet as a Global Platform for Biomedical Research." *Nature Health*, vol. 1, 2026, pp. 792- 795. 10.1038/s44360-026-00096-z.

XVI. Sharma, S., et al. "Design and Implementation of Portable ECG Monitoring System Using ESP32 and Web-Based Visualization." *International Journal of Creative Research Thoughts*, vol. 13, 2025, pp. 34-82.

XVII. Virtanen, P., et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." *Nature Methods*, vol. 17, no. 3, 2020, pp. 261-72. 10.1038/s41592-019-0686-2.

XVIII. Xu, Z., et al. "Multimodality Data Augmentation Network for Arrhythmia Classification." *International Journal of Intelligent Systems*, vol. 2024, 2024, article 9954821, pp. 1-14. 10.1155/2024/9954821.

XIX. Zhou, F., and D. Fang. "Multimodal ECG Heartbeat Classification Method Based on a Convolutional Neural Network Embedded with FCA." *Scientific Reports*, vol. 14, no. 1, 2024, article 59311, pp. 1-10. 10.1038/s41598-024-59311-0.

XX. Zishan, M. A., et al. "Dense Neural Network Based Arrhythmia Classification on Low-Cost and Low-Compute Micro-Controller." *Expert Systems with Applications*, vol. 239, 2024, article 122560, pp. 1-12. 10.1016/j.eswa.2023.122560.