

AI-Powered Holter for Affordable and Accurate Arrhythmia Detection

Nyatte Steyve ¹, Guiadem Leatitia¹, Perabi steve^{1,2} and Ndjakomo Essiane^{1,2}

¹ Laboratory of Technology and Applied Sciences, University of Douala (Douala, Cameroon)

² Signal, Image and Systems Laboratory, University of Ebolowa (Ebolowa, Cameroon)

ABSTRACT

Cardiac arrhythmias pose significant health risks, and current detection systems often suffer from high costs and limited accessibility, particularly in resource-constrained settings. This research aimed to develop a portable, cost-effective Holter monitoring device for accurate arrhythmia detection using machine learning. By combining an inexpensive ESP32 microcontroller with an AD8232 ECG sensor, a data acquisition system was built. Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Multilayer Perceptron (MLP) models were trained and evaluated for arrhythmia classification. The SVM model achieved the highest accuracy (78.53%) using a linear kernel and features selected by a random forest algorithm. While KNN and MLP also showed promise, the results emphasized the importance of hyperparameter tuning and feature selection. This research demonstrated the feasibility of creating an affordable and intelligent Holter device capable of effective arrhythmia detection, potentially increasing access to cardiac monitoring and enabling early diagnosis in resource-limited environments.

PAPER HISTORY

Received Feb. 26, 2025

Revised Mei 30, 2025

Accepted 13 June, 2025

Published June 27, 2025

KEYWORDS

Cardiac Arrhythmia Detection;
Holter Device;
ESP32 Microcontroller; Machine Learning;
Cost-Effective Healthcare.

CONTACT:

ssteyve@gmail.com

ngoffeperabi@yahoo.fr

guiademleatitia@gmail.com

salomendjakomo@gmail.com

1. INTRODUCTION

Cardiac arrhythmias, characterized by irregular heartbeats, pose significant health risks, including stroke, heart failure, and sudden cardiac arrest. The prevalence of these conditions necessitates effective monitoring systems, particularly for high-risk populations. However, the existing cardiac monitoring solutions, such as traditional Holter monitors, often come with high costs and require specialized medical personnel for operation and interpretation [1]. This limits their accessibility, especially in low-resource settings where healthcare infrastructure is inadequate [2]. As a result, many patients remain undiagnosed or receive delayed treatment, highlighting the urgent need for affordable and user-friendly arrhythmia detection solutions [3]. Several methods have been employed to detect cardiac arrhythmias, each with its advantages and disadvantages. For instance, deep learning techniques, such as convolutional neural networks (CNNs), have shown high accuracy in ECG classification [4]. However, these models require substantial computational resources and large datasets, which may not be available in all settings. Traditional machine learning algorithms, such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN), offer more accessible alternatives but often necessitate careful feature extraction and selection [5]. Additionally, wearable devices equipped with sensors, although convenient, may suffer from inaccurate data due to motion artifacts [6]. Furthermore, many existing solutions depend on proprietary software, limiting their adaptability and increasing costs [7]. Cardiac arrhythmias pose significant

health risks and are often undiagnosed due to the limitations of the existing detection systems, particularly in resource-limited environments. The need for accessible and cost-effective solutions has led to various innovative approaches in the development of Holter monitoring devices.

Ranjha et al. (2023) [8] explored the design of a wireless Holter monitor utilizing neural networks for ECG analysis. Their approach involved developing a low-cost cardiac screening system that integrates a wireless ECG module for real-time signal acquisition, with data transmitted to a cloud server. The results demonstrated a classification accuracy exceeding 88%. However, the implementation in resource-limited settings remains a challenge, indicating a gap that necessitates further exploration. Building upon this concept, Huda et al. (2023) [9] developed a portable, energy-efficient ECG system that incorporates cloud-based arrhythmia detection. Their device utilizes an AD8232 circuit for ECG signal acquisition, transmitting data via Bluetooth for analysis. This system achieved a remarkable accuracy of 94.03% in classifying abnormal heart rhythms. The advancements in portable technology emphasize the potential for widespread application but also underscore the importance of balancing cost with performance. In a comparative study, Hyun et al. (2024) [10] evaluated a portable ECG device (MobiCARE-MC100) against a traditional 24-hour Holter monitor for detecting atrial fibrillation post-cardiac surgery. Both devices identified an equivalent number of paroxysmal A-fib cases within the initial 24 hours. Notably, a follow-up with MobiCARE

revealed an increased A-fib burden in one patient (from 9% to 38%). While MobiCARE demonstrated comparable accuracy, the limited sample size poses constraints on the generalizability of these findings. Huang et al. (2024) [11] investigated the effectiveness of a two-hour AI-based Holter monitoring system for detecting premature ventricular contractions (PVCs) and supraventricular contractions (PVCs). Their study aimed to assess whether this method could outperform traditional 24-hour monitoring. Among 170 patients, the two-hour Holter displayed a positive predictive value (PPV) of 76.00% and a negative predictive value (NPV) of 96.35%, compared to 87.50% and 98.55%, respectively, for the 24-hour device. However, the limited sample size again raises questions about the robustness of the results.

Yazid et al. (2023) [12] proposed a lightweight method for detecting atrial fibrillation from electrocardiographic signals using an improved dynamic threshold algorithm. By reducing the input feature size to 44 without sacrificing accuracy, they achieved impressive performance metrics: sensitivity of 99.14%, specificity of 99.12%, and accuracy of 99.13% over 15-second segments. For 60-second signals, these values reached 99.49%, 99.46%, and 99.47%, respectively. Implemented on an Arm Cortex M4 microcontroller, the method operates at a mere 27 mA, although its algorithmic complexity might hinder its application in highly constrained devices. Finally, Patibandla et al. (2024) [13] compared 24-hour ECG monitoring between a traditional Holter and the Vigo Heart patch across 119 patients. The Vigo Heart exhibited a significantly lower average noise percentage (1.94% vs. 17.84% for Holter). They also observed meaningful correlations in maximum (129.69 vs. 113.31 bpm), average (74.85 vs. 76 bpm), and minimum (47.94 vs. 59.21 bpm) heart rates. The consistency of cardiologist diagnoses across participants suggests that the Vigo Heart presents a viable alternative for ambulatory arrhythmia monitoring.

The reviewed studies highlight the ongoing efforts to enhance the accessibility and effectiveness of Holter monitoring devices for cardiac arrhythmia detection. While advancements in wireless technology, portable systems, and innovative algorithms show promise, challenges remain in terms of cost, implementation, and generalizability. This literature underscores the necessity for further research to optimize these systems, paving the way for improved patient care and early arrhythmia detection, particularly in resource-limited settings. The present study aims to contribute to this body of knowledge by developing a cost-effective and intelligent Holter device that leverages machine learning algorithms for accurate arrhythmia detection. Despite the advancements in arrhythmia detection technology [14–18], significant gaps remain in the development of affordable [19–21], portable, and user-friendly systems [22–24]. Most existing studies focus on complex algorithms that demand significant computational power, which can be prohibitive for widespread use, particularly in developing regions. Additionally, there is a lack of integration of commonly available hardware components, such as microcontrollers, with effective machine learning

algorithms. This research aims to bridge these gaps by leveraging accessible technology to create a practical solution for cardiac monitoring.

The primary aim of this study is to design and develop a low-cost, intelligent Holter device that utilizes an ESP32 microcontroller and machine learning algorithms for accurate cardiac arrhythmia detection. By focusing on easily obtainable components and avoiding the complexities of deep learning, this research seeks to provide a viable solution for continuous cardiac monitoring. This approach does not only enhance accessibility but also encourages the adoption of such technology in resource-limited environments, ultimately improving health outcomes for patients at risk of arrhythmias. This research contributes to the field by demonstrating the feasibility and efficacy of a simplified, cost-effective approach to arrhythmia detection, potentially paving the way for wider deployment of such life-saving technology, particularly in underserved communities.

2. MATERIALS AND METHOD

The detection of cardiac arrhythmias from electrocardiogram (ECG) data is a critical area in cardiology. Traditional methods rely heavily on visual interpretation of ECGs by experts, a process that is time-consuming, costly, and prone to inter-observer variability. The application of machine learning models and algorithms provides a promising alternative to automate this process, enhance diagnostic accuracy, and increase access to care. The development of a portable Holter device for cardiac arrhythmia detection necessitates a thorough understanding of both the hardware and the underlying mathematical modeling for energy consumption and machine learning algorithms. This section outlines the key theoretical concepts that support the design, functionality, and efficiency of the device.

A. Energy Consumption and Battery Sizing

To ensure continuous operation for 24 hours, the Holter system must be powered efficiently. Our design operates on a 5V power supply, and it is crucial to accurately estimate the total energy consumption of all components involved. The following measurements were taken from the prototype :

- ESP32-WROOM-32: Consumed 150 mA at 3.3V (approximately 0.5W). Converted to 5V with 90% efficiency yields a consumption of about 112 mA.
- AD8232 Amplifier Module: Consumed 10 mA at 5V (0.05W).
- SD Card Module: Consumed 5 mA at 3.3V (approximately 0.0165W). At 5V, with 90% efficiency, this equates to about 3.6 mA.
- 2.4-inch TFT LCD Display: Consumed 100 mA at 5V (0.5W).
- Other Components (peripheral circuits, etc.): Consumed 20 mA at 5V (0.1W).

To find the total energy consumption over 24 hours, we use equation (1):

$$\begin{aligned} \text{TotalEnergyConsumption}(Wh) &= \left(\frac{245.6}{1000}\right) * 5 * 24h \\ &= 2.96Wh \end{aligned} \quad (1)$$

To determine the required battery capacity, we must account for overall efficiency during charging and discharging, assumed to be 80%. The necessary battery capacity must exceed the total energy consumption divided by the efficiency, as shown in equation (2):

$$\begin{aligned} \text{Battery Capacity}(Wh) &= \frac{\text{Total Energy Consumption}(Wh)}{\text{Efficiency}} \\ &= \frac{2.96}{0.8} = 3.6 \end{aligned} \quad (2)$$

For a LiPo battery rated at 5V, the required capacity in mAh can be calculated using equation (3):

$$\text{Capacity}(mAh) = \frac{\text{Capacity}(Wh) * 1000}{\text{Voltage}(V)} = \frac{3.96 * 1000}{5} = 997mAh \quad (3)$$

To ensure reliable operation, it is advisable to select a rechargeable LiPo battery with a capacity of at least 1000 mAh. Considering potential variations in consumption, a battery rated at 1200 mAh or more would be preferable to enhance longevity and ensure consistent performance. The machine learning models implemented in this study SVM, KNN, and MLP—rely on mathematical foundations for classification tasks. The performance of these algorithms is influenced by the choice of features extracted from the ECG signals [25-27].

1. SVM Mathematic model

The mathematical modeling of a Support Vector Machine (SVM) can be described in several ways, depending on whether we consider the linearly separable or non-linear case.

2. Linearly Separable Case :

In this case, the goal is to find the optimal hyperplane that separates the two classes with the maximum margin. This hyperplane is defined by the equation (4):

$$w^T x + b = 0 \quad (4)$$

For instance, SVM employs the concept of maximizing the margin between classes by solving a constrained optimization problem, which can be mathematically represented as follows in equation (5) [23]:

$$\text{Minimize } \frac{1}{2} \|w\|^2 \text{ subject to } y_i(w^T x_i + b) \geq 1, \forall i \quad (5)$$

Here, w is the weight vector, b is the bias, and y_i are the labels of the training data x_i . This optimization problem can be solved using the method of Lagrange multipliers, which leads to the dual formulation (6):

$$\text{Maximize } \sum \alpha_i - \left(\frac{1}{2}\right) \sum \sum \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (6)$$

subject to Eq(7) and (8):

$$0 \leq \alpha_i \leq C \text{ for all } i \quad (7)$$

$$\sum \alpha_i y_i = 0 \quad (8)$$

where:

α_i are the Lagrange multipliers.

C is a regularization parameter that controls the trade-off between margin maximization and classification error minimization.

3. Non-linearly Separable Case:

In this case, we used a kernel function to project the data into a higher-dimensional space where it can be linearly separated. The optimization problem then became the following equation (9) :

$$\text{Maximize } \sum \alpha_i - \left(\frac{1}{2}\right) \sum \sum \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (9)$$

subject to (7) and (8)

where $K(x_i, x_j)$ is the kernel function that calculates the dot product of the feature vectors projected into the higher-dimensional space. Common examples of kernel functions include:

Linear Kernel: $K(x_i, x_j) = x_i^T x_j$

Polynomial Kernel: $K(x_i, x_j) = (x_i^T x_j + c)^d$

Gaussian (RBF) Kernel: $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$

The choice of kernel function and its parameters is crucial for the performance of the SVM.

In summary, the mathematical modeling of an SVM is based on optimizing an objective function that aims to maximize the margin between classes, either in the original feature space or in a higher-dimensional space through the use of a kernel function.

d) KNN math modelling

KNN, on the other hand, relies on calculating distances in feature space [23]. The classification decision is based on the majority voting mechanism among the 'k' nearest neighbors. K-Nearest Neighbors (KNN) operates on the principle that similar data points tend to share the same class. It classifies a new data point by considering the majority class among its 'k' nearest neighbors. The algorithm's effectiveness relies on the chosen distance metric and the value of 'k'.

- Feature Representation:** Each data point is represented as a vector, $x = (x_1, x_2, \dots, x_d)$, in a d-dimensional feature space, where x_i represents the i-th feature's value.
- Distance Metric:** The Minkowski distance, a generalized metric encompassing both Euclidean and Manhattan distances, is employed (equation 10) :

$$d(x, y) = (\sum_i |x_i - y_i|^p)^{\frac{1}{p}} = \|x - y\|_p \quad (10)$$

Here, 'p' is a parameter:

- $p = 1$ corresponds to Manhattan distance:
 $d(x, y) = \sum_i |x_i - y_i|$
- $p = 2$ corresponds to Euclidean distance:
 $d(x, y) = \sqrt{(\sum_i (x_i - y_i)^2)}$

Finding the K-Nearest Neighbors: For a new data point, we computed its Minkowski distance to all training data points. The 'k' data points with the smallest distances were selected as its nearest neighbors.

Classification: The new data point was assigned to the majority class among its k-nearest neighbors. Tie-

breaking strategies include random selection or distance-weighted voting. While not a strict mathematical formula, the classification can be represented as:

$$\text{Class}(x_{\text{new}}) = \arg\max_c \sum_i I(\text{Class}(x_{\text{neighbor}_i}) = c) \quad (11)$$

where:

- x_{new} is the new data point.
- x_{neighbor_i} is the i -th nearest neighbor.
- $\text{Class}(x)$ denotes the class of data point x .
- $I(\text{condition})$ is an indicator function (1 if true, 0 otherwise).
- $\arg\max_c$ returns the class ' c ' maximizing the sum.

This effectively counts neighbors per class, assigning the new point to the most frequent class among its neighbors. The choice of ' p ' in the Minkowski distance and the value of ' k ' are critical parameters influencing KNN's performance and are tuned based on the specific dataset and problem.

a) MLP Mathematical Modeling within the Holter Design

This section details the mathematical underpinnings of the Multilayer Perceptron (MLP) used in our Holter monitor design. The MLP was chosen for its ability to learn complex non-linear relationships within the ECG data.

- **Neuron Model:** Each neuron in the MLP transforms its inputs as follows (eq 12):

$$\text{output} = \text{ReLU}(\sum_i (w_i x_i + b)) \quad (12)$$

We utilized the Rectified Linear Unit (ReLU) activation function for its computational efficiency and effectiveness in deep learning: $\text{ReLU}(z) = \max(0, z)$.

Here, x_i represents the i -th input, w_i its corresponding weight, and b the neuron's bias.

- **Network Architecture and Calculation:** Our MLP employed a single hidden layer, striking a balance between complexity and computational cost. The output of the hidden layer (a^h) and the final output layer (a^o) are calculated as equation 13 and 14:

$$a^h = \text{ReLU}(W^h a^i + b^h) \quad (13)$$

$$a^o = \text{sigmoid}(W^o a^h + b^o) \quad (14)$$

where:

a^i is the input vector.

W^h and b^h are the weight matrix and bias vector for the hidden layer, respectively.

W^o and b^o are the weight matrix and bias vector for the output layer, respectively.

$\text{sigmoid}(z) = 1 / (1 + \exp(-z))$ is used in the output layer for binary classification, producing a probability score.

- **Loss Function and Optimization:** We trained the MLP using the binary cross-entropy loss function show by equation 15:

$$\text{Loss} = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \quad (15)$$

where y is the true label (0 or 1) and \hat{y} is the predicted probability. This loss function is suitable for our binary classification task. We employed the Adam optimizer, a variant of stochastic gradient descent, to minimize the loss and update the network's weights and biases during training.

MLPs utilized feedforward neural networks, where the output was derived from the weighted sum of inputs passed through activation functions [24 25]. The input layers in the model were fully connected to the hidden layer and the output layer is fully connected to the hidden layer.

The theoretical background provided in this section demonstrates the foundational principles governing the design and functionality of the Holter device. By thoroughly understanding energy consumption and machine learning modeling, this study aims to develop an efficient and effective system for continuous cardiac monitoring, addressing the crucial need for accessible healthcare technology.

B. Dataset Description

The dataset employed for training in this study is the "Arrhythmia" dataset from the UCI Machine Learning Repository, which comprises data from 452 patients, characterized by 279 attributes

(<http://archive.ics.uci.edu/ml/datasets/Arrhythmia>). These attributes vary in type, including integer, real, and categorical values. The primary objective is to classify patients into 16 distinct groups: one group representing normal ECG readings (class 01) and the remaining groups representing different types of arrhythmias (classes 02 to 15). A final category (class 16) is designated for unclassified cases.

C. DATA ACQUISITION

The Holter monitor system's architecture prioritizes low-cost, energy-efficient components capable of acquiring, processing, displaying, and storing ECG data for a minimum of 24 hours. Figure 1 illustrates the system's interconnection and the roles of each component.

The data acquisition process began with the ECG Sensor, which captured the electrical activity of the heart. This sensor was connected to a Conditioning Circuit responsible for amplifying and filtering the raw ECG signal, removing noise and artifacts to prepare it for digital processing. The conditioned signal was then fed into the central processing unit, an ESP32 microcontroller. The ESP32 is a low-power, Wi-Fi-enabled microcontroller ideal for wearable applications like our Holter monitor. It performs the crucial task of digitizing the analog ECG signal and executing the machine learning algorithms for arrhythmia detection. The ESP32 interacts with two key storage components. Firstly, an SD Card provides non-volatile storage for the recorded ECG data, allowing for long-term data logging and subsequent analysis. Secondly, the ESP32's internal memory is used for temporary storage and buffering during real-time processing. The system also includes a TFT Screen to display real-time ECG waveforms and any detected arrhythmia alerts to the user. Finally, the entire system is

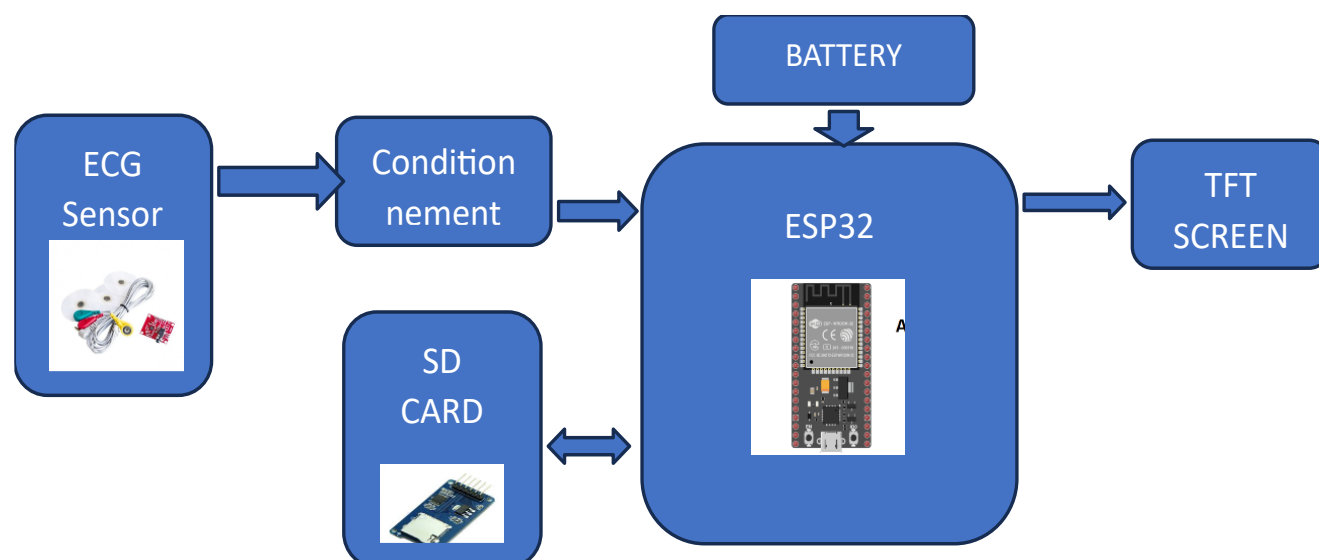


Fig. 1. Synopsis of the system

powered by a Battery, selected for its capacity to sustain operation for the required 24-hour monitoring period. The system's design emphasizes minimizing power consumption to maximize battery life. The data acquisition system designed for this study integrates various hardware components that work together to capture, process, and transmit ECG signals for arrhythmia detection. The key components of the system include the ESP32 microcontroller, the AD8232 ECG sensor, an SD card module for data storage, and a display module for user interaction. This section outlines the functionality and roles of each component in the overall data acquisition system.

The ESP32 microcontroller serves as the central processing unit of the data acquisition system. It is a low-cost, low-power microcontroller with integrated Wi-Fi and Bluetooth capabilities, making it an ideal choice for portable applications. The ESP32 is responsible for: It processes the raw ECG signals obtained from the AD8232 sensor, applying necessary filtering and feature extraction algorithms to enhance signal quality. Microcontroller interfaces with an SD card module to store the acquired ECG data for further analysis and model training. It manages communication between the ECG sensor, storage module, and any user interface components, ensuring seamless data flow within the system and implement the AI algorithm.

The AD8232 ECG sensor is a critical component for acquiring the electrical signals generated by the heart. It is designed for low-power, low-noise applications and is capable of providing high-fidelity ECG signals. Key features of the AD8232 include:

Signal Acquisition: The sensor captures the ECG signals through electrodes placed on the patient's skin. It amplifies these signals while filtering out noise and interference, providing a clean output suitable for further processing.

Analog Output: The AD8232 outputs an analog signal that represents the heart's electrical activity, which is then

digitized by the ESP32's analog-to-digital converter (ADC) for processing.

To facilitate data storage, the system incorporates an SD card module that allows the ESP32 to write ECG data in real-time. This module is essential for:

Data Logging: It stores large volumes of ECG data, enabling long-term monitoring of patients without immediate data transfer requirements.

Post-Processing: The stored data can be accessed later for detailed analysis and algorithm training, ensuring that valuable information is not lost during continuous monitoring.

A 2.4-inch TFT LCD is integrated into the system to provide real-time feedback to users. This component is useful for:

User Interface : It allows healthcare professionals or patients to visualize the ECG data in real-time, improving user engagement and facilitating immediate feedback.

Alerts and Notifications : The display show alerts for detected arrhythmias, ensuring timely intervention if necessary.

The data acquisition process follows a systematic flow :

1. **Signal Capture :** The AD8232 sensor captures the heart's electrical signals through electrodes placed on the skin.
2. **Signal Amplification and Filtering:** The sensor amplifies and filters the raw signals to reduce noise.
3. **Analog-to-Digital Conversion :** The ESP32 converts the analog signals from the AD8232 into digital format for processing.
4. **Data Storage :** The processed ECG data is logged onto the SD card for future analysis.
5. **Data analysis to make prediction**
6. **Real-Time Display :** the display module shows the ECG signals and alerts for any detected abnormal patterns.

The data acquisition system, comprising the ESP32 microcontroller, AD8232 ECG sensor, SD card module, and optional display, forms a cohesive unit designed to capture and process ECG signals efficiently. By integrating these components, the system facilitates the reliable detection and prediction of cardiac arrhythmias, contributing to improved patient monitoring and care.

D. DATA PROCESSING

Data processing is a crucial step in the development of a reliable cardiac arrhythmia detection system. This phase encompasses several key activities, including filtering, feature extraction, and the application of machine learning algorithms. Each of these processes contributes to transforming raw ECG signals into meaningful data that can be effectively analyzed for arrhythmia classification. For filtering the raw ECG signals, bandpass filtering was chosen as the most effective method. This technique allows signals within a specific frequency range to pass through while attenuating frequencies outside this range. For ECG signals, a bandpass filter is typically set between 0.5 Hz and 40 Hz. This range effectively removes high-frequency noise (such as muscle artifacts) and low-frequency drift (such as baseline wander). The mathematical representation of a bandpass filter can be expressed in the frequency domain as follows:

$$H(f) = \begin{cases} 1, & \text{for } f_1 < f < f_2 \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

where $H(f)$ is the filter's transfer function, and f_1 and f_2 are the cutoff frequencies. From the main database consisting of 279 features, we proceed to select the most

important parameters. To do this, we used two methods to determine which one yields the best results: PCA (Principal Component Analysis) and Random Forest. We employed a Random Forest classifier for feature selection, capitalizing on its inherent ability to rank features based on their contribution to predictive accuracy. This approach allows for a data-driven selection of the most informative features, improving model efficiency and mitigating overfitting. Specifically, we utilized the `feature_importances_` attribute of the trained `sklearn.ensemble.RandomForestClassifier` (Python's scikit-learn library). This attribute provides a score for each feature, reflecting its importance in the model's decision-making process. Higher scores indicate greater predictive power. Our feature selection process followed these steps:

1. **Initial Feature Set:** We began with a comprehensive set of 279 features extracted from the ECG signals via Continuous Wavelet Transform (CWT). This included diverse time-frequency characteristics and morphological features derived from the QRSTP complexes.
2. **Random Forest Training:** A Random Forest classifier was trained on the dataset using all 279 features and their corresponding arrhythmia labels.
3. **Feature Importance Ranking:** The `feature_importances_` attribute was extracted from the trained model, providing a ranked list of features based on their importance scores.
4. **Threshold Determination:** We applied a threshold to select the most relevant features. This threshold was determined empirically by analyzing the

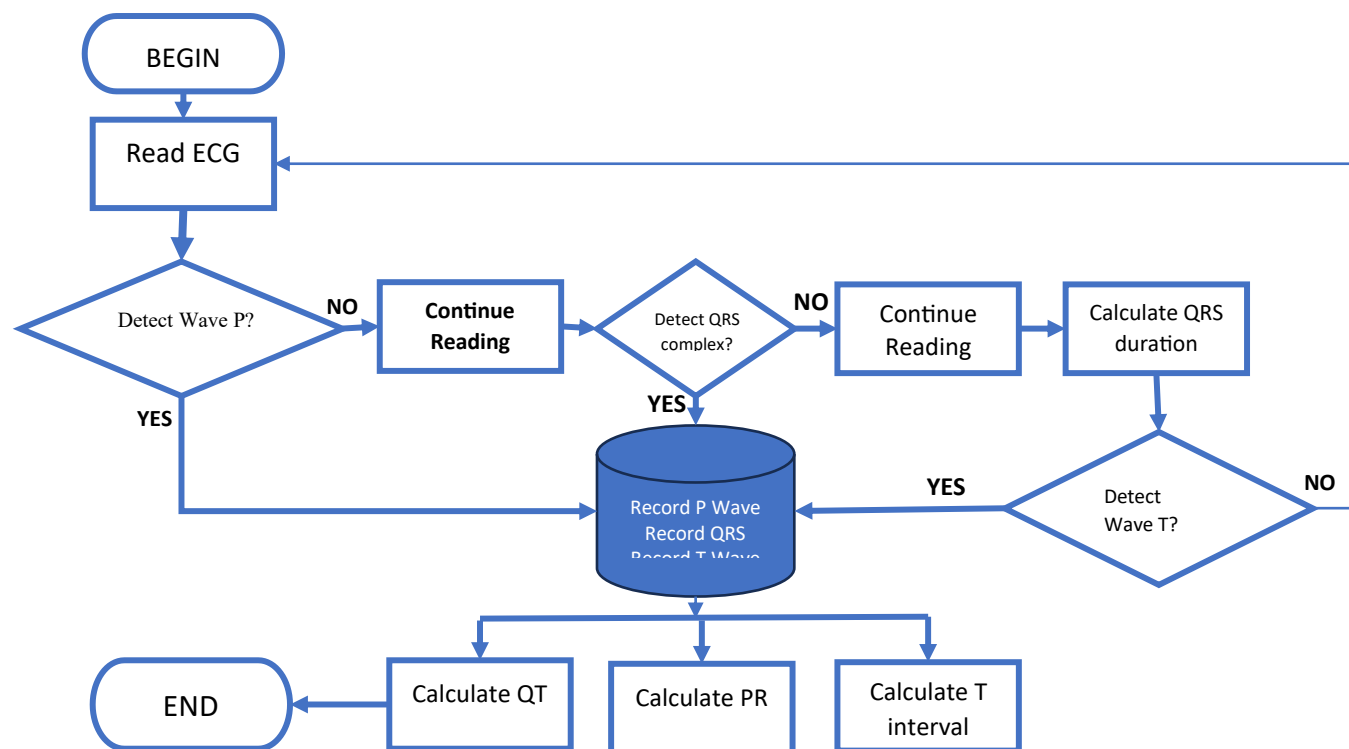


Fig. 2. Features extraction flowchart

distribution of importance scores. Our goal was to retain features contributing significantly while discarding less informative ones. We opted for a data-driven approach, selecting the top N features that collectively accounted for 95% of the total importance score.

5. **Final Feature Subset:** The features whose importance scores surpassed the defined threshold constituted the final, reduced feature subset used for training the arrhythmia detection model.

This method offers several advantages. It leverages the Random Forest's ability to assess feature importance in a complex, non-linear manner. The 95% cumulative importance threshold ensures the retention of the most impactful features while reducing dimensionality. It does not only improve computational efficiency but also reduce the risk of overfitting by eliminating less relevant or redundant features, ultimately enhancing the model's generalizability and predictive performance. To analyze and extract the QRSTP characteristics from the acquired ECG signal, the wavelet transform method was selected due to its ability to analyze non-stationary signals like ECG comprehensively. Wavelet analysis provides a multi-resolution approach, allowing for simultaneous examination of both time and frequency characteristics. This is particularly useful for detecting transient features in ECG signals that are indicative of arrhythmias. The Continuous Wavelet Transform (CWT) can be mathematically expressed as:

$$W(a, b) = \int_{-\infty}^{+\infty} x(t) \Psi^* \left(\frac{t-b}{a} \right) dt \quad (17)$$

Where Ψ is the wavelet function a is the scale parameter, and b is the translation parameter. This method enables the extraction of significant features such as QRSTP complexes, which are crucial for accurate arrhythmia classification. By employing bandpass filtering and wavelet transform for filtering and feature extraction, the data processing pipeline for the cardiac arrhythmia detection system is optimized to enhance signal quality and extract relevant characteristics from ECG data. The flowchart in Figure 2 details the algorithm for ECG signal processing and feature extraction within the Holter system. This process is crucial for identifying key components of the ECG waveform and calculating relevant intervals, which were then used as features for machine learning-based arrhythmia detection.

The algorithm began by reading the ECG signal ("Read ECG"). It then proceeds to detect the P-wave ("Detect P Wave?"). If a P-wave is detected ("YES"), the algorithm continues to search for the QRS complex ("Detect QRS Complex?"). If no P-wave is detected ("No"), the algorithm continues reading the ECG signal ("Continue Reading") until a P-wave is found. Similarly, if a QRS complex is detected ("YES"), the algorithm proceeds to calculate the QRS duration ("Calculate QRS Duration") and subsequently searches for the T-wave ("Detect T Wave?"). If no QRS complex is detected ("No"), the algorithm continues reading the ECG signal ("Continue Reading").

Once the P-wave, QRS complex, and T-wave are identified and recorded ("Record P Wave," "Record QRS,"

"Record T Wave"), the algorithm calculates crucial intervals: the QT interval ("Calculate QT Interval"), the PR interval ("Calculate PR"), and the T interval ("Calculate T Interval"). These calculated intervals, along with the QRS duration, serve as essential features for characterizing the ECG signal and are subsequently used by the machine learning models for classification. If no T wave is detected ("No"), the algorithm continues reading the ECG signal ("Continue Reading") to find a T wave. Once all features are extracted for a given heartbeat, the process repeats for the next heartbeat until the end of the monitoring period ("End"). This sequential process ensures that all relevant features were extracted from each heartbeat, providing a comprehensive dataset for accurate arrhythmia detection. The "Continue Reading" steps in the flowchart ensure that the algorithm continues processing the ECG signal even if a particular wave or complex is not immediately detected, making the system robust to noise and variations in ECG morphology.

These techniques ensure that the system can effectively analyze and classify arrhythmias, contributing to improved patient monitoring and care. The data analysis phase of the cardiac arrhythmia detection study focuses on evaluating the performance of the machine learning models using metrics that specifically assess classification effectiveness. This section outlines the statistical methods and performance metrics that will be employed, including accuracy, precision, and F1 score as shown by equations 18, 19, 20, 21:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (18)$$

$$Precision = \frac{TP}{TP + FP} \quad (19)$$

$$Recall = \frac{TP}{TP + FN} \quad (20)$$

$$F1 - score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad (21)$$

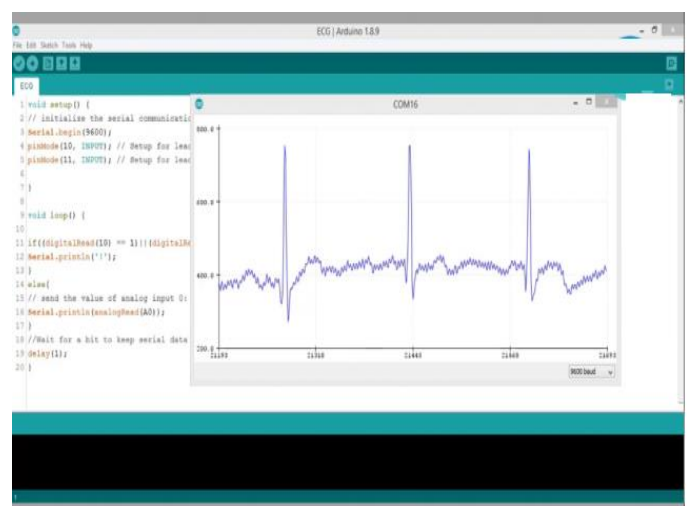


Figure 3: The ECG curve displayed on Processing

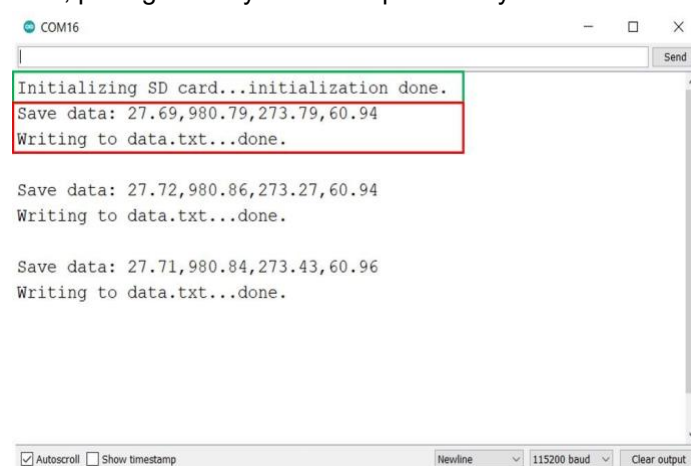
3. RESULTS

A. ECG Signal Detection

After assembling and testing the circuit on a breadboard, it was clear from Figure 3 that the ECG signal was

accurately detected. The various waves, such as the P wave, QRS complex, and T wave, were distinctly visible, allowing for easy determination of the heart rate. This curve shows the ECG curve displayed on Processing. This curve illustrates a successfully captured and processed cardiac signal via an Arduino board. The image indicates clear identification of characteristics such as P waves, QRS complexes, and T waves, which are distinct elements of a normal ECG signal. The regularity of the signal suggests a steady heart rhythm, facilitating easy measurement of heart rate. However, it is important to note that this visual observation is preliminary. A more in-depth quantitative analysis, including measurements of heart rate, RR intervals, and assessment of noise or artifacts, is necessary for a comprehensive validation of signal quality and acquisition system accuracy.

Figure 4 illustrates a screenshot from a serial terminal software, demonstrating the process of recording ECG data onto an SD card. The initial step indicates successful initialization of the SD card ("Initializing SD card...initialization done."). Following this, multiple lines of data are displayed, each starting with "Save data:" followed by a series of comma-separated numerical values. These values represent the acquired ECG data, measuring various parameters (amplitude, duration, etc.) at different time points. Each line concludes with "Writing to data.txt...done," confirming the successful writing of data to a file named "data.txt" on the SD card. The repeated display of "Save data:" and "Writing to data.txt...done" suggests a continuous acquisition and recording process of the ECG data. The serial transmission speed was set to 115200 baud, indicating successful acquisition and storage of ECG data on the SD card, paving the way for subsequent analysis.



```

COM16
Initializing SD card...initialization done.
Save data: 27.69,980.79,273.79,60.94
Writing to data.txt...done.

Save data: 27.72,980.86,273.27,60.94
Writing to data.txt...done.

Save data: 27.71,980.84,273.43,60.96
Writing to data.txt...done.
  
```

Figure 4: The process of recording ECG data onto an SD card.

B. Validation of the Artificial Intelligence Model

1. Database

The MIT-BIH Arrhythmia Database used for training and validating the model comprises 10,942 instances and 28 attributes. Two feature selections methods were utilized: the first derived from Principal Component Analysis (PCA), encompassing 88 principal components explaining of the total variance, and the second derived from feature selection using random forests, comprising

the most important features according to the Gini criterion..

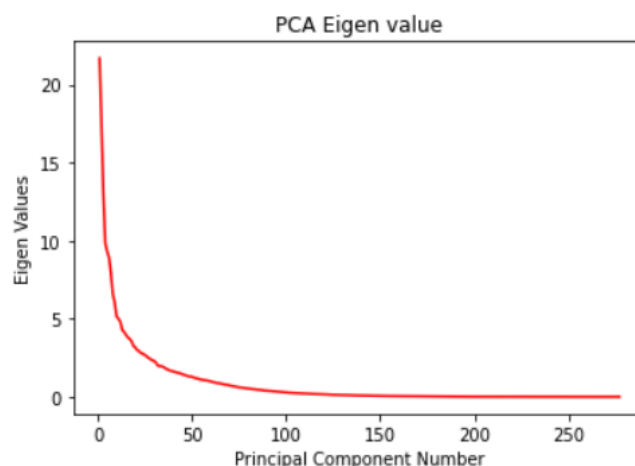


Figure 5: Eigenvalues Obtained After Applying PCA

The figure 5 presents a graph of the eigenvalues obtained after applying PCA to the ECG data. The x-axis represents the principal component number, while the y-axis represents the corresponding eigenvalue. A rapid decline in eigenvalues was observed at the beginning of the graph, indicating that the first principal components capture a significant portion of the data variance. The decline slows down thereafter, suggesting that subsequent principal components contribute less to the total variance. This elbow-shaped curve helps determine the optimal number of principal components to retain. Retaining the first 88 principal components, as mentioned earlier, was justified by their representation of a significant proportion of total variance while considerably reducing data dimensionality, thus simplifying the model and enhancing computational efficiency. The elbow observed around the 88th principal component supports this choice. The features are move to 14 et The best instances were age, sex; height, weight, QRSduration, PRinterval, Q-Tinterval, Tinterval, Pinterval, QRS, T, P, QRST, heartrate and classfeatures.

2. Data Processing

Before training our machine learning model, we preprocessed the ECG data to ensure optimal performance. Specifically, we applied Min-Max normalization using `sklearn.preprocessing.MinMaxScaler` to scale the relevant features to a range between 0 and 1. This normalization technique enhances the convergence speed of the training algorithms and improves the overall robustness of the model. We selected Min-Max scaling after comparing its performance with other scaling methods.

Table 1: Min-Max Normalization via `sklearn.preprocessing.MinMaxScaler`:

QT	PR	RR	ST
0.152631	0.155123	0.445066	0.562732
58	34	25	.	47

0.082775 12	0.092504 74	0.450112 98	0.562416 05
.....
0.139712 92	0.152277 04	0.472616 68	0.583765 43

Table 1 illustrates the effect of Min-Max normalization on several key ECG features: QT interval, PR interval, RR interval, and ST segment. Each row in the table represents a different sample or instance from the dataset. The columns display the normalized values of the corresponding features after applying the Min-Max scaler. For example, the first row shows that the original QT, PR, RR, and ST values have been transformed to 0.15263158, 0.15512334, 0.44506625, and 0.56273247, respectively. The "..." entries in the table and the RR column represent additional features or data points not shown for brevity but treated similarly during the normalization process. This normalization ensures that all features contribute equally to the model's learning process, regardless of their original scales or ranges.

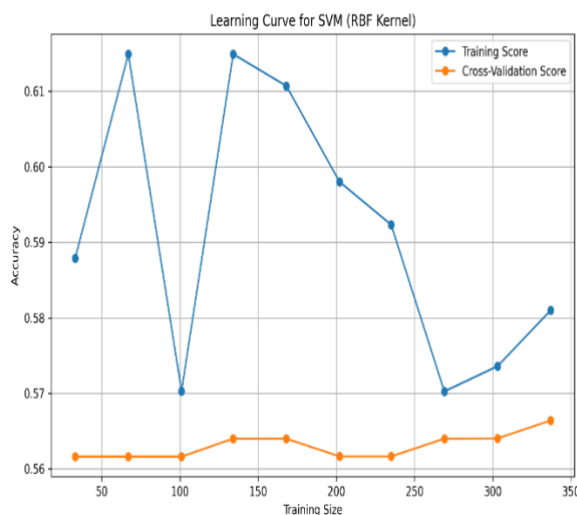


Figure 6: The Impact of Kernel Choice and Critical Factor

3. Model Training

This section presents the results of classification experiments conducted using three algorithms (SVM, KNN, MLP) on the preprocessed ECG data for us to choose the best algorithm in term of processing time and metrics. The classifiers' performance was evaluated using precision, recall, F1 score, and confusion matrix on an independent test set. Figure 6 depicts the learning curve for the SVM model employing an RBF kernel, illustrating performance as a function of training set size. The x-axis represents the training set size, while the y-axis represents accuracy. The blue line indicates the training score, reflecting performance on the training data. The orange line represents the cross-validation score, a more robust measure of generalization performance on unseen data. To ensure robust model evaluation and address potential overfitting, we provide further details on our data

handling procedures. The dataset, derived from the MIT-BIH Arrhythmia Database, comprised 10,942 instances and was reduced to 14 key features (as detailed in the Dataset Description section). A stratified 70/30 split was employed to create training and testing sets, preserving the class distribution across both. The reported performance metrics were calculated based on the held-out test set, which remained untouched during model training and hyperparameter optimization. Hyperparameter tuning was performed systematically using a grid search combined with 5-fold cross-validation. The regularization parameter (C) was explored over a range of 0.1 to 100 on a logarithmic scale. For each C value, 5-fold cross-validation was performed, and the C value maximizing average cross-validation accuracy was selected. Figure 6 reveals that both training and cross-validation scores generally increase with training set size, indicating the model's capacity to learn from additional data. The consistently higher training score compared to the cross-validation score was expected, reflecting optimization on the training data. The gap between these scores suggests some overfitting, particularly with smaller training sets. Increasing training set size leads to improved cross-validation scores, signifying better generalization. The converging trend of both curves suggests potential further performance gains with additional data. This learning curve analysis supports the robustness of the hosen SVM model with an RBF kernel for arrhythmia detection.

Table 2: Classification report for Linear SVM

	precision	recall	f1-score	support
1	0.61	0.95	0.74	46
2	0.5	0.11	0.18	9
3	0	0	0	4
4	0	0	0	3
5	0.5	0.33	0.4	3
6	0.66	0.66	0.66	3
7	0	0	0	0
8	0	0	0	1
10	1	0.3	0.333333	10
14	0	0	0	1
16	0	0	0	5
accuracy	0.588235	0.588235	0.588235	0.588235
macro avg	0.29798	0.206148	0.211598	85
weighted avg	0.542484	0.588235	0.499703	85

For the SVM model, various kernels were tested: linear, polynomial, Gaussian (RBF), and sigmoid. The linear kernel provided the best performance, achieving an accuracy of 78.53% as shown in figure 6. The figure 7 presents the results of 7-fold cross-validation applied to a

K-nearest neighbors (KNN) classifier for classifying cardiac arrhythmias.:

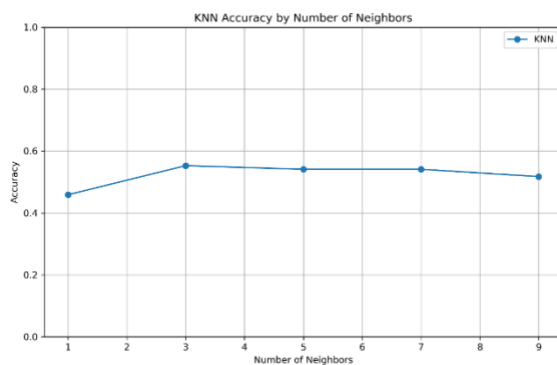


Figure 7: 7-Fold Cross-Validation Applied to K-Nearest Neighbors Classifier

The 7-fold cross-validation was performed on a KNN model with 5 neighbors ($k=5$) using features extracted by PCA. The average accuracy obtained is 61.26%. The associated graph shows the accuracy trend based on the number of neighbors (k) tested. Similarly, a 7-fold cross-validation was performed on a KNN model with 3 neighbors ($k=3$) using features selected by random forest. The average accuracy obtained is 65.51%. The associated graph illustrates the accuracy trend based on the number of neighbors (k) tested.

The results demonstrate that the choice of the number of neighbors (k) influences the accuracy of the KNN model. For both feature selection methods (PCA and random forest), there exists a k value that optimizes accuracy. Additionally, using features selected by random forest leads to better overall accuracy (65.51%) compared to using features derived from PCA (61.26%). This suggests that random forest feature selection identified a more relevant subset of features for classifying cardiac arrhythmias with the KNN classifier. The graphs illustrate the accuracy variation for different k values, enabling the identification of the optimal k value for each feature selection method.

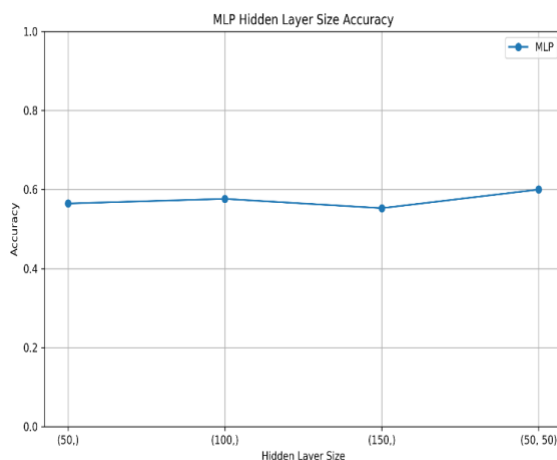


Figure 8 Results from a Multilayer Perceptron (MLP) Model

Figure 8 presents the results of a multilayer perceptron (MLP) model applied to the classification of cardiac arrhythmias. Several observations can be made:

Overall Performance: The model achieves an accuracy of 66.91% on the test set and 68.63% on the validation set. The slight difference between these two values suggests a minor overfitting issue, but overall, the model generalizes reasonably well.

Table 3: Classification report for MLP

1	0.554217	1	0.713178	46
2	1	0.111111	0.2	9
3	0	0	0	4
4	0	0	0	3
5	0	0	0	3
6	1	0.333333	0.5	3
8	0	0	0	1
10	0	0	0	10
14	0	0	0	1
16	0	0	0	5
accuracy	0.564706	0.564706	0.564706	0.564706
macro avg	0.255422	0.144444	0.141318	85
weighted avg	0.441106	0.564706	0.424779	85

The MLP model achieves generally acceptable performance, but the classification report (table 3) reveals significant disparities between classes. While some classes are classified well, others pose challenges for the model. Further analysis of classification errors and the characteristics of misclassified classes is necessary to enhance model performance.

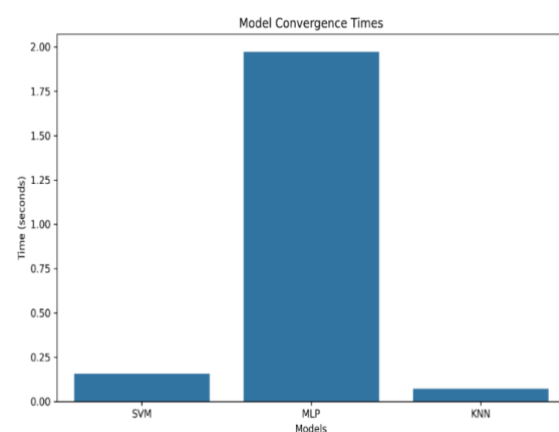


Figure 9: Models convergence times

4. Discussion

A. Analysis of Results

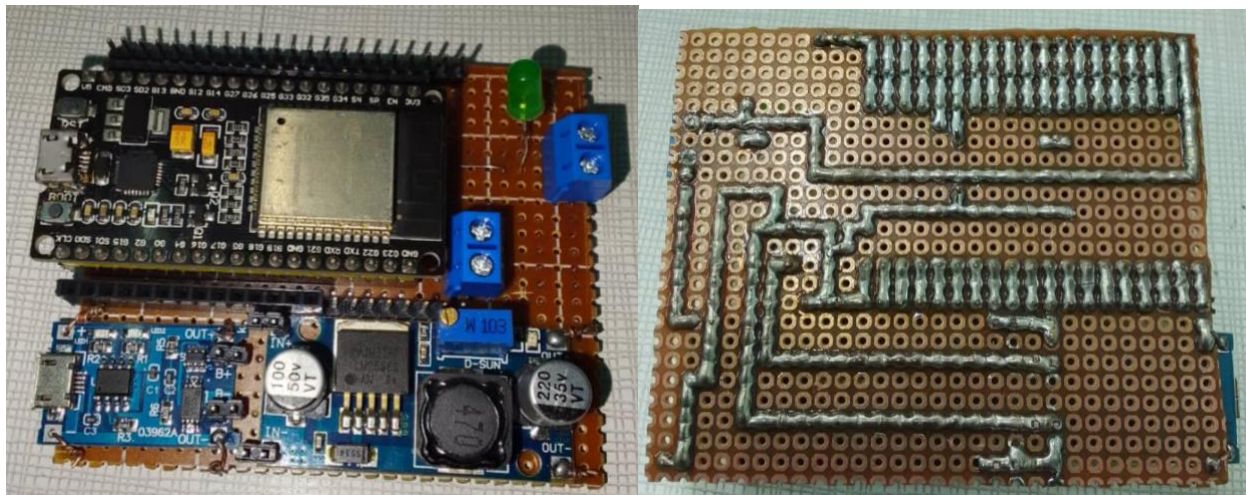


Figure 10. Electronic circuit of the Holter



Figure 11: 3D printing model of the Holter

The comparative analysis of the three AI models (SVM, KNN, and MLP) for classifying cardiac arrhythmias highlights the strengths and weaknesses of each approach. The SVM model displayed notable sensitivity to the choice of kernel and the critical factor (regularization parameter). With an accuracy of 78.53%, the linear kernel outperformed others, suggesting a relatively simple linear relationship between features and arrhythmia classes. However, the performance of the RBF and polynomial kernels indicates the presence of non-linear relationships in the data, which warrants further exploration. Utilizing feature selection via random forests slightly improved the SVM's performance compared to PCA, underscoring the importance of appropriate feature selection. Optimizing the critical factor is crucial to avoid both overfitting and underfitting. The KNN model demonstrated significant dependence on the choice of the number of neighbors (k). The 7-fold cross-validation revealed better performance with random forest feature

selection (65.51%) compared to PCA (61.26%). This reinforces the importance of feature quality in maximizing model performance. Identifying the optimal k value is essential for enhancing the accuracy of the KNN model. The MLP model achieved an accuracy of 66.91% on the test set and 68.63% on the validation set. While it shows acceptable performance, significant disparities in performance across classes indicate that some arrhythmias were misclassified. This highlights the need for a detailed analysis of classification errors and potential improvements in network architecture or the use of data augmentation techniques. The comparative analysis of the three AI models (SVM, KNN, and MLP) for classifying cardiac arrhythmias reveals the strengths and weaknesses of each approach, with important implications for clinical application. While the SVM with a linear kernel achieved the highest accuracy (78.53%), this metric alone does not fully capture the clinical impact. In a real-world setting, the consequences of false positives

(incorrectly classifying a normal heartbeat as an arrhythmia) and false negatives (failing to detect a true arrhythmia) can be significantly different. A high rate of false positives could lead to unnecessary anxiety, additional testing, and potentially even inappropriate interventions. Conversely, false negatives could delay diagnosis and treatment, potentially with serious health consequences. Therefore, optimizing sensitivity (the ability to correctly identify true positives) and specificity (the ability to correctly identify true negatives) is crucial for clinical applicability. Future work will focus on strategies to improve these metrics, such as exploring cost-sensitive learning approaches that penalize false negatives more heavily.

The bar chart in figure 9 provides a clear comparison of the time taken for each machine learning model to converge. the convergence time analysis indicates that the SVM model is a highly efficient choice for our Holter cardiac monitoring system, particularly when rapid decision-making is crucial. The integration of the arrhythmia detection system within the smart garment relies on biomedical sensors connected to an ESP32 board. First, we designed and 3D printed the casing for the smart Holter monitor, as illustrated in the figure 10. The Holter monitor's electronic circuit, depicted in Figure 11, was implemented on a perforated board using soldering techniques. The circuit integrates several key components: an ESP32 microcontroller, an AD8232 ECG sensor, a power management module, and supporting circuitry. The ESP32 serves as the central processing unit, responsible for data acquisition, signal processing, and communication. The AD8232 sensor captures the ECG signals, which were then processed by the ESP32. The power management module provides a stable voltage supply to the various components, ensuring consistent operation. The supporting circuitry includes components like resistors, capacitors, and LEDs, which play essential roles in signal conditioning, filtering, and visual feedback. The perforated board provides a flexible platform for prototyping and allows for easy modification and debugging of the circuit. The soldering method ensures secure connections between the components,

enhancing the reliability and durability of the device. This implementation approach balances functionality, cost-effectiveness, and ease of assembly, making it suitable for a portable and accessible Holter monitor.

Figure 12 illustrates the Holter monitor in operation, showcasing its ability to capture and display cardiac signals in real-time. The displayed ECG waveform on the integrated screen provides immediate visual feedback, allowing for on-the-spot monitoring of heart activity. This feature is particularly useful for initial assessment and verification of proper device functionality. The compact and portable design of the Holter, evident in the figure, emphasizes its suitability for ambulatory monitoring. The following table 4 presents a comparison of various cardiac monitoring studies, highlighting their methodologies, accuracy, key metrics, and notable insights. While our proposed Holter system using an SVM with an RBF kernel demonstrates promising results with a 78.33% accuracy, it's crucial to acknowledge its limitations and potential weaknesses in comparison to the existing approaches. One key limitation is the observed variability in performance with different kernel types. Further investigation and optimization of kernel selection are necessary to ensure consistent and robust performance.



Figure 12: the Holter designed

Table 4. Comparison of various cardiac monitoring studies

Study/Model	Methodology	Accuracy (%)	Other Metrics	Notes
Ranjha et al. (2023) [8]	Wireless Holter monitor with neural networks	88	Real-time signal acquisition, cloud integration	The prediction is performed in the cloud. This does not solve the problem of insufficient computing resources.
Huang et al. (2024) [11]	AI-based two-hour Holter monitoring	N/A	PPV: 76.00%, NPV: 96.35%	Aimed to outperform traditional monitoring; questioned robustness due to limited sample size.
Our Model (SVM with RBF)	Support Vector Machine with RBF kernel	78.33	Variability with kernel types	AI is integrated directly into the designed device rather than relying on cloud-based solutions as noted in [8]

Comparing our approach to the existing studies reveals important distinctions. Ranjha et al. (2023) [8] achieved a higher accuracy of 88% with their neural network-based wireless Holter monitor. However, their system relies on cloud integration for prediction, which presents a significant drawback. Cloud dependency introduces latency, requires a constant internet connection, and raises privacy concerns regarding sensitive patient data. Our system addresses this limitation by integrating the AI directly into the device, enabling on-device processing and eliminating the need for cloud connectivity. Huang et al. (2024) [11] focused on AI-based two-hour Holter monitoring, achieving a Positive Predictive Value (PPV) of 76.00% and a Negative Predictive Value (NPV) of 96.35%. While their study aimed to improve upon traditional Holter monitoring, the authors themselves acknowledge concerns about the robustness of their findings due to a limited sample size. Furthermore, their study has shorter monitoring duration (two hours) limits its applicability for detecting less frequent arrhythmias that might manifest over longer periods. Our system, designed for 24-hour monitoring, offers a more comprehensive approach to arrhythmia detection.

In summary, while our proposed method may not achieve the highest reported accuracy compared to cloud-based solutions, its on-device processing capability offers significant advantages in terms of real-time analysis, reduced latency, enhanced privacy, and independence from internet connectivity. Future work will focus on refining kernel selection and exploring other model optimization strategies to further improve accuracy and robustness. Direct comparison with studies like Huang et al. (2024) [11] is challenging due to differences in reported metrics and study durations, highlighting the need for standardized evaluation protocols in this field.

4. CONCLUSION

The primary objective of this study was to develop an integrated smart garment capable of detecting cardiac arrhythmias using advanced machine learning algorithms. By leveraging physiological data from biomedical sensors and implementing effective signal processing techniques, the system aims to provide accurate and timely detection of arrhythmias, thereby enhancing patient monitoring and care. The results of the comparative analysis of three machine learning models Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Multilayer Perceptron (MLP) revealed distinct strengths and weaknesses in each approach. The SVM model with a linear kernel achieved the highest accuracy of 78.53%, indicating a relatively simple linear relationship between features and arrhythmia classes. The KNN model demonstrated performance sensitivity to the choice of the number of neighbors, while the MLP model exhibited acceptable accuracy but struggled with class imbalances. Additionally, the integration of feature selection through random forests enhanced the performance of both the SVM and KNN models. Overall, the smart garment

system shows considerable potential for early arrhythmia detection, although challenges related to false positives and class imbalances remain. Future research will focus on several key areas to enhance the effectiveness of the arrhythmia detection system. Efforts will be made to optimize the signal processing algorithms further, incorporating advanced noise reduction techniques and exploring additional feature extraction methods. Additionally, the integration of data from other sensors, such as accelerometers and respiratory monitors, will be pursued to enrich the model and improve diagnostic accuracy. Investigating more sophisticated machine learning techniques, including convolutional and recurrent neural networks, will also be part of the future work. Finally, large-scale clinical validation involving diverse patient populations and arrhythmia types will be essential to confirm the system's robustness and reliability in real-world applications. Enhancing the garment's comfort and usability, alongside developing a user-friendly interface for data visualization and interpretation, will be crucial for patient acceptance and engagement.

REFERENCES

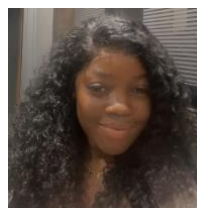
- [1]. Knox, M., Huang, B., Willard-Grace, R., & Su, G. (2024, March). Innovation in public delivery systems: How one safety net hospital implemented new heart monitoring technology. In *Healthcare* (Vol. 12, No. 1, p. 100732). Elsevier.
- [2]. Li, Q., Kanduma, E., Ramiro, I., Xu, D., Cuco, R. M. M., Chaquise, E., ... & Pan, J. (2024). Spatial access to continuous maternal and perinatal health care services in low-resource settings: cross-sectional study. *JMIR Public Health and Surveillance*, 10, e49367.
- [3]. Sridhar, A. R., Cheung, J. W., Lampert, R., Silva, J. N., Gopinathannair, R., Sotomonte, J. C., ... & Lakkireddy, D. (2024). State of the art of mobile health technologies use in clinical arrhythmia care. *Communications Medicine*, 4(1), 218.
- [4]. Sahoo, S., Dash, M., Behera, S., & Sabut, S. (2020). Machine learning approach to detect cardiac arrhythmias in ECG signals: A survey. *Irbm*, 41(4), 185-194.
- [5]. Udurume, M., Shakhov, V., & Koo, I. (2024). Comparative Analysis of Deep Convolutional Neural Network—Bidirectional Long Short-Term Memory and Machine Learning Methods in Intrusion Detection Systems. *Applied Sciences*, 14(16), 6967.
- [6]. Prabakaran, A., & Rufus, E. (2022). Review on the wearable health-care monitoring system with robust motion artifacts reduction techniques. *Sensor Review*, 42(1), 19-38.
- [7]. Muhammad, T. (2022). A Comprehensive Study on Software-Defined Load Balancers: Architectural Flexibility & Application Service Delivery in On-Premises Ecosystems. *International Journal of Computer Science and Technology*, 6(1), 1-24.
- [8]. A. Ranjha, L. Jabbar, et O. Ahmed, « Cloud-Connected Wireless Holter Monitor Machine with

- Neural Networks Based ECG Analysis for Remote Health Monitoring », 21 octobre 2023, *arXiv*: arXiv:2310.13965. doi: 10.48550/arXiv.2310.13965.
- [9]. N. Huda, S. Khan, R. Abid, S. B. Shuvo, M. M. Labib, et T. Hasan, « A Low-cost, Low-energy Wearable ECG System with Cloud-Based Arrhythmia Detection », in *2020 IEEE Region 10 Symposium (TENSYP)*, juin 2020, p. 1840-1843. doi: 10.1109/TENSYP50017.2020.9230619.
- [10]. S. Hyun, S. Lee, Y. S. Hong, S. Lim, et D. J. Kim, « Evaluation of the Diagnostic Performance and Efficacy of Wearable Electrocardiogram Monitoring for Arrhythmia Detection after Cardiac Surgery », *J. Chest Surg.*, vol. 57, n° 2, p. 205-212, mars 2024, doi: 10.5090/jcs.23.152.
- [11]. Q. Huang *et al.*, « The diagnostic efficiency of artificial intelligence based 2 hours Holter monitoring in premature ventricular and supraventricular contractions detection », *Clin. Cardiol.*, vol. 47, n° 4, p. e24266, 2024, doi: 10.1002/clc.24266.
- [12]. M. Yazid, M. A. Rahman, N. Nuryani, et Aripriharta, « Atrial Fibrillation Detection From Electrocardiogram Signal on Low Power Microcontroller », *IEEE Access*, vol. 12, p. 91590-91604, 2024, doi: 10.1109/ACCESS.2024.3422329.
- [13]. S. Patibandla, K. Kumar, R. Adepu, R. K. Bandaru, et B. Maduri, « The Validation of a Mobile Based Ambulatory Heart Rhythm Monitoring Solution - Vigo Heart. | EBSCOhost ». Consulté le: 29 mars 2025. [En ligne]. Disponible sur: https://openurl.ebsco.com/EPDB%3Agcd%3A13%3A4867821/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A177979222&cl=c&link_origin=scholar.google.fr
- [14]. Wang, Y. C., Xu, X., Hajra, A., Apple, S., Kharawala, A., Duarte, G., ... & Faillace, R. T. (2022). Current advancement in diagnosing atrial fibrillation by utilizing wearable devices and artificial intelligence: A review study. *Diagnostics*, 12(3), 689.
- [15]. Guess, M., Zavanelli, N., & Yeo, W. H. (2022). Recent advances in materials and flexible sensors for arrhythmia detection. *Materials*, 15(3), 724.
- [16]. Liu, J., Li, Z., Jin, Y., Liu, Y., Liu, C., Zhao, L., & Chen, X. (2022). A review of arrhythmia detection based on electrocardiogram with artificial intelligence. *Expert review of medical devices*, 19(7), 549-560.
- [17]. Ansari, Y., Mourad, O., Qaraqe, K., & Serpedin, E. (2023). Deep learning for ECG Arrhythmia detection and classification: an overview of progress for period 2017–2023. *Frontiers in Physiology*, 14, 1246746.
- [18]. Hambly, H., & Rajabiun, R. (2021). Rural broadband: Gaps, maps and challenges. *Telematics and Informatics*, 60, 101565.
- [19]. Aranda, M. P., Kremer, I. N., Hinton, L., Zissimopoulos, J., Whitmer, R. A., Hummel, C. H., ... & Fabius, C. (2021). Impact of dementia: Health disparities, population trends, care interventions, and economic costs. *Journal of the American Geriatrics Society*, 69(7), 1774-1783.
- [20]. Kim, N. W., Joyner, S. C., Riegelhuth, A., & Kim, Y. (2021, June). Accessible visualization: Design space, opportunities, and challenges. In *Computer graphics forum* (Vol. 40, No. 3, pp. 173-188).
- [21]. Dong, G., Chen, B., Liu, B., Hounjet, L. J., Cao, Y., Stoyanov, S. R., ... & Zhang, B. (2022). Advanced oxidation processes in microreactors for water and wastewater treatment: Development, challenges, and opportunities. *Water Research*, 211, 118047.
- [22]. Aranda, M. P., Kremer, I. N., Hinton, L., Zissimopoulos, J., Whitmer, R. A., Hummel, C. H., ... & Fabius, C. (2021). Impact of dementia: Health disparities, population trends, care interventions, and economic costs. *Journal of the American Geriatrics Society*, 69(7), 1774-1783.
- [23]. Randazzo, V., Ferretti, J., & Pasero, E. (2021). Anytime ECG monitoring through the use of a low-cost, user-friendly, wearable device. *Sensors*, 21(18), 6036.
- [24]. Ehresh, M., Abatis, P., & Schlindwein, F. S. (2020). A portable electrocardiogram for real-time monitoring of cardiac signals. *SN Applied Sciences*, 2(8), 1419.
- [25]. Priya, E., & Chitra, R. (2021, December). Smartphone based portable ECG monitoring system. In *2021 4th International Conference on Computing and Communications Technologies (ICCT)* (pp. 121-126). IEEE.

AUTHOR BIOGRAPHY



Dr Steyve Nyatte received a Ph.D. degree in Artificial Intelligence from the University of Douala. He is currently a Lecturer with the University Institute of Technology of Douala and a member of the ESE research team. His research interests include biomedical signal processing, Optimization in medical diagnosis, machine learning, and Neglect tropical diseases.



Guiadem Leatitia is a Master's research student at the University of Douala, working in the Laboratory of Technologies and Applied Sciences. Holding a Master of Technology degree in Biomedical Engineering, she specializes in the field, particularly in optimizing artificial intelligence for low-cost devices.



Pr STEVE PERABI NGOFFE is a renowned expert in electrical engineering and control systems at the University of Douala in Cameroon. His skills and expertise encompass a wide range of areas, including hybrid systems, photovoltaics, material modeling, electrical power engineering,



Pr Salomé Ndjakomo Essiane is a prominent figure in the field of electrical engineering and renewable energies in Cameroon. She currently holds the positions of Director of HTTTC (Higher Technical Teacher Training College) and Head of Department at the University of

Douala. Her expertise and skills include electrical engineering, power systems simulation, MATLAB

simulation, electrical & electronics engineering, renewable energy technologies, power electronics, power systems modeling, and harmonics. She has actively published more than 75 research articles on ResearchGate, demonstrating her significant contributions to the field. You can reach her at salomendjakomo@gmail.com. Pr Ndjakomo Essiane's expertise, experience, and research activity make her a valuable asset to the development of the energy sector in Cameroon

Affordable Arrhythmia Detection System

