Adaptive Model Selection for Real-Time Heart Disease Detection on Embedded Systems

Yixin Li*, Zhiling Li†, Abdullah Al Arafat*, Donald Johnson‡, Ning Sui*, Anil Gehi§, Zhishan Guo*
*North Carolina State University, †Southern University of Science and Technology
‡Evergreen Healthcare, §UNC School of Medicine
{yli223, aalaraf, nsui, zguo32}@ncsu.edu, lizl2021@mail.sustech.edu.cn, donald.johnson@evergreenhc.net, anil_gehi@med.unc.edu

***Abstract*—Real-time cardiovascular disease (CVD) detection on wearable devices presents significant challenges due to the varying heart rate conditions and constrained computational capabilities of embedded systems. Existing approaches often struggle to balance diagnostic accuracy with the strict latency requirements imposed by different heart rate scenarios. In this study, we propose an Adaptive Model Selection (AMS) framework coupled with an anytime Convolutional Neural Network that integrates Residual Blocks, Squeeze-and-Excitation layers, and a Global Attention mechanism. By dynamically adjusting the model's complexity based on real-time heart rate, our solution optimizes diagnostic accuracy while ensuring a timely response. Evaluations conducted with the PhysioNet Database on a Raspberry Pi 4 demonstrate that our model achieves an accuracy of 91.5% with an average inference latency of only 1.33 ms per sample. These outcomes illustrate the effectiveness and practical applicability of our framework for robust, responsive, and accurate on-device ECG monitoring in continuous cardiac care. Our code is available online on GitHub[1].**

***Index Terms*—electrocardiography monitoring, real-time embedded system, adaptive model selection, deep learning**

I. Introduction

Cardiovascular diseases (CVDs) remain the leading cause of mortality globally, accounting for approximately 17.9 million deaths annually [21]. Sudden cardiac events can occur unpredictably, even among individuals with no prior heart condition diagnoses, highlighting the critical need for continuous cardiac monitoring for the general population [5], [18]. Medical-grade monitoring devices, although accurate, are typically expensive, inconvenient for daily use, and thus not widely adopted by the general public. Consequently, consumer-grade wellness devices such as smartwatches and fitness trackers, which integrate ECG monitoring functionalities, have gained significant popularity for everyday health management.

Wearable ECG devices offer continuous, non-invasive cardiac monitoring by leveraging cyber-physical system capabilities and human-in-the-loop interactions, enabling personalized health insights and timely interventions [22], [30]. However, real-time ECG processing on these devices poses unique challenges due to dynamically varying heart rates. Traditional fixed-window ECG analysis methods [15], [16], although commonly used, fail to accommodate variations in heart rate effectively. When using a shifted window approach for data segmentation, a higher heart rate implies more cardiac cycles within a given fixed-size window, thereby requiring increased computational effort. For instance, a single heartbeat cycle necessitates only one inference, whereas three cycles demand three separate computations. Such variability can lead to missed deadlines and compromised responsiveness, especially under higher heart rate conditions.

Furthermore, consumer wearable devices often handle multiple concurrent tasks, including user interface management, wireless communication (e.g., Bluetooth, Wi-Fi), sensor fusion processes, data encryption, and logging [6]. These concurrent tasks further constrain computational resources, complicating real-time ECG anomaly detection and necessitating adaptive resource demand strategies based on heart rate variability.

To address this critical challenge, we propose an Adaptive Model Selection (AMS) framework, complemented by an anytime convolutional neural network. Our approach dynamically selects model complexity based on real-time heart rate monitoring. Specifically, at higher heart rates, a lightweight model ensures timely computation, while moderate and advanced models provide deeper, more accurate analyses at lower heart rates. By dynamically adjusting model complexity according to instantaneous physiological conditions, our solution effectively balances computational efficiency with diagnostic accuracy, enhancing reliability and responsiveness in cardiac monitoring applications.

Integrating advanced signal processing techniques, adaptive machine learning methodologies, and dynamic real-time scheduling, our proposed framework significantly improves the feasibility and accuracy of ECG monitoring on resource-constrained wearable devices, promoting broader adoption for continuous cardiac care.

Contributions. Our contributions are summarized as follows:

- **Novel ECG Detection Model:** An advanced convolutional neural network architecture combining Residual Blocks, Squeeze-and-Excitation (SE) layers, and Global Attention mechanisms to enhance diagnostic accuracy.
- **Adaptive Model Selection (AMS) Framework:** A dynamic selection strategy among three CNN variants—advanced, moderate, and lightweight—tailored to real-time computational resource availability in wearable devices.
- **Unified Anytime Model:** A single, parameter-shared model architecture with early-exit points to adapt seam-

[1] https://github.com/yixinli19/AMS_CVD

2325-1301/25/$31.00 ©2025 IEEE
DOI 10.1109/RTCSA66114.2025.00028

lessly to varying computational constraints, significantly reducing memory usage on embedded platforms.

- **Comprehensive Evaluation:** Extensive experiments using the PhysioNet 2021 Challenge dataset on a Raspberry Pi 4, benchmarking against state-of-the-art models, and demonstrating superior accuracy and real-time responsiveness.

Paper Organization. We begin by outlining the research problem in Section II. Section III details the real-time system design, while Section IV further explains the schedulability analysis of the real-time system. Section V elaborates the deep learning model design on CVD detection. Section VI discusses the adaptive model selection and anytime model. In addition, Section VII presents the experiments and results, followed by a discussion in Section VIII. Section IX reviews related literature, and Section X concludes the study and suggests directions for future works.

II. Problem Statement

Wearable ECG devices are critical for continuous, real-time monitoring of cardiovascular health by capturing the heart's electrical signals. However, achieving accurate and timely anomaly detection on these embedded devices is challenging due to their constrained computational resources and inherent variability in heart rate (HR).

Heart rate HR, defined as the number of heartbeats per minute, significantly influences the computational load in ECG analysis. As shown in Fig. 1, higher HR results in more frequent ECG cycles within a fixed timeframe, increasing computational demands as the comparatively larger number of ECG segments (defined as a window with a fixed number of ECG cycles) need to analyze than a HR yielding fewer cycles.

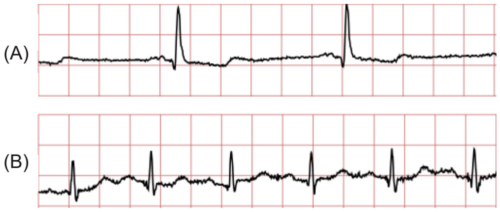


Fig. 1: A 3-second ECG recording. In Part A, the R-R interval spans 5 large grid boxes (1.0 second), corresponding to a heart rate of 60 beats per minute. In Part B, the R-R interval spans 2 large grid boxes (0.4 seconds), corresponding to a heart rate of 150 beats per minute.

To effectively manage the variation of computational demand of ECG signal analysis, the system dynamically selects the appropriate anomaly detection model complexity based solely on instantaneous heart rate. The selection of suitable model complexity at runtime ensures that diagnostic accuracy remains high while real-time processing deadlines, dictated by heart rate, are consistently met. Thus, the core challenge lies in developing an adaptive framework capable of selecting the optimal anomaly detection model in real-time, considering variability in HR as the primary influencing factor.

Our objective is to design an adaptive cardiovascular disease anomaly detection system that maximizes diagnostic accuracy while meeting real-time constraints. During runtime, to process i^{th} ECG segment E^i, an optimal model $M_i \in \{M_{\text{Adv}}, M_{\text{Mod}}, M_{\text{LW}}\}$, representing advanced, moderate, and lightweight models, respectively, is selected to maximize diagnostic accuracy meeting deadline constraints.

Formally, the optimization problem is defined as follows:

$$\begin{aligned} \underset{M_i}{\arg\max} \quad & A(M_i) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{I}\left(M_i(E^i) = y_i\right) \\ \text{s.t.} \quad & T_{\text{proc}}(M_i) \le D(\text{HR}), \\ & M_i \in \{M_{\text{Adv}},\, M_{\text{Mod}},\, M_{\text{LW}}\} \end{aligned} \tag{1}$$

where $\mathbf{I}(.)$ is an indicator function, and y_i and $M_i(E^i)$ denote the ground-truth and model-predicted labels, respectively. Here, the diagnostic accuracy $A(M_i)$ measures the proportion of correctly classified ECG samples out of the total N samples. The constraint $T_{\text{proc}}(M_i) \le D(\text{HR})$ ensures real-time responsiveness by adjusting the processing deadline based on the instantaneous HR. The performance of M_i depends on the complexity level of the model; for example, an advanced model would yield better accuracy than a moderate or lightweight model but would require more execution time.

By solving this optimization problem at each iteration, the system adaptively selects the most suitable anomaly detection model, effectively balancing high diagnostic accuracy and real-time constraints amid changing heart rate conditions.

III. System Design

The proposed system processes single-lead ECG data in real-time, ensuring accurate anomaly detection across varying heart rates. The workflow, depicted in Fig. 2, includes the key stages of the proposed method.

ECG data is continuously captured using a single-lead configuration. To enable efficient real-time processing, a shifted window method segments the ECG signal into overlapping, fixed-duration windows. This method ensures continuous data analysis with minimal latency.

Within each window, R-peak detection is performed using Hamilton's algorithm [10], accurately identifying individual heartbeats. Precise detection of R-peaks facilitates extracting heartbeat cycles, defined as intervals between consecutive R-peaks (R-R intervals). Example segments containing varying numbers of cardiac cycles are illustrated in Fig. 3.

Our system dynamically adapts to the number of cardiac cycles within each fixed-duration window. If a window contains more cycles due to a higher heart rate, the system correspondingly performs more computational tasks, as each detected cardiac cycle triggers an inference execution.

Following R-peak and cycle detection, the system dynamically selects the most suitable model complexity based on the current heart rate:

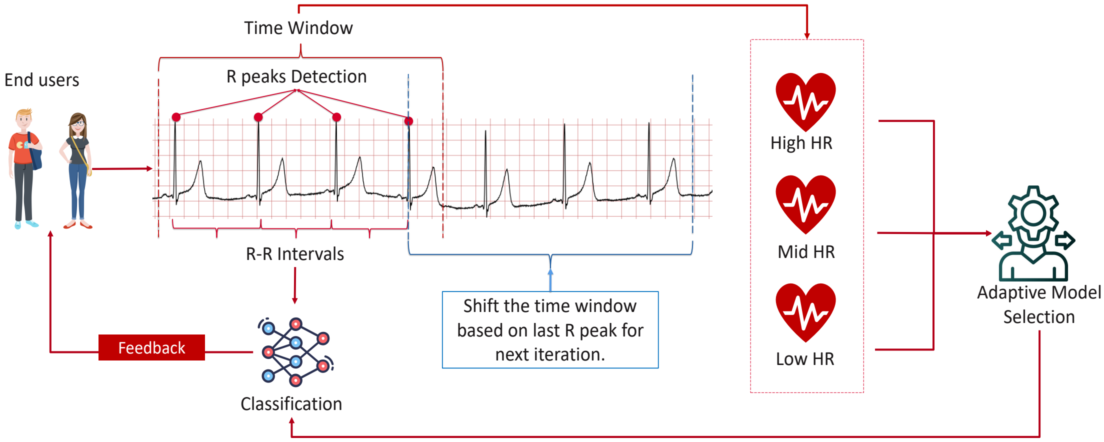


Fig. 2: Workflow of the real-time ECG processing system, including data acquisition, shifted window segmentation, R-peak detection, heart rate monitoring, adaptive model selection based on heart rate conditions, and user alerting for abnormal detection.

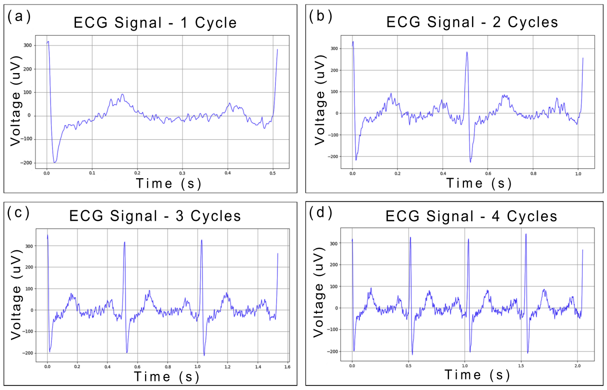


Fig. 3: Representative ECG signal segments with varying numbers of cardiac cycles. Each subfigure shows a continuous ECG waveform consisting of (a) one, (b) two, (c) three, and (d) four cardiac cycles, respectively.

- **High** HR (HR $\geq$ 90): Employ a lightweight model to rapidly handle multiple cycles within constrained deadlines.
- **Moderate** HR (70 $\leq$ HR $<$ 90): Use a moderate-complexity model for balanced performance.
- **Low** HR (HR < 70): Utilize a comprehensive model, taking advantage of additional processing time to maximize diagnostic accuracy.

Once processed with the selected model, ECG data is evaluated for anomalies. If cardiac activity is deemed normal, monitoring continues seamlessly. If abnormalities are detected, the system immediately alerts the user, ensuring timely medical intervention and enhancing safety.

This adaptive and dynamic cycle-based approach effectively addresses varying computational demands imposed by different heart rates, balancing real-time constraints with diagnostic accuracy, thereby providing robust ECG monitoring suitable for resource-constrained wearable devices.

IV. Schedulability Analysis

We consider a set of recurrent workload $\Gamma = \{E, \tau_1, \tau_2, \ldots, \tau_n\}$ is scheduled on a uniprocessor system. Here, E is an adaptive varying task that presents the ECG segment analyzed for anomaly detection. E can potentially have infinite number of instance and i^{th} instance of E is denoted as E^i. Each ECG segment is denoted by $(C(\text{HR}), D(\text{HR}), \text{HR})$ where, $C(\text{HR})$ is the worst-case execution time, $D(\text{HR})$ is the relative deadline and also sampling period, i.e., consecutive samples are at least $D(\text{HR})$ unit apart, and HR is the user's heart rate during sampling instant of E. Both $C(\text{HR})$ and $D(\text{HR})$ are the functions of HR. Fig. 4 illustrates the relation between HR and $D(\text{HR})$. $C(\text{HR})$ includes worst-case execution time to process E by anomaly detection model M. For instance, E^i can be processed by M_{Mod} based on instantaneous heart rate HR, but the E^{i+1} can be processed by M_{LW} implying that $C^i(\text{HR}) > C^{i+1}(\text{HR})$. The remaining tasks of Γ, $\{\tau_1, \tau_2, \ldots, \tau_n\}$, are sporadic tasks to model all other processes in the system, where each task τ_i is modeled as (C_i, T_i). Here, C_i is the worst-case execution time, and T_i is the period (also the relative deadline of the task). Each of these tasks can also potentially release infinite instances with a minimum inter-arrival time of T_i.

The tasks in the system are scheduled preemptively using the Earliest Deadline First (EDF) scheduling algorithm. Under (work-conserving) EDF scheduling, the task instance with the earliest absolute deadline among all eligible tasks is always selected for execution on the uniprocessor.

The schedulability of a task set with relative deadlines equal to periods, scheduled under EDF on a uniprocessor, can be effectively analyzed using the Liu and Layland schedulability

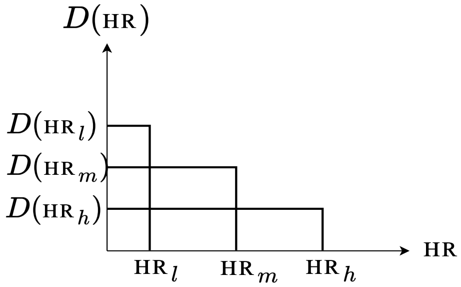


Fig. 4: Relationship between heart rate (HR) and relative deadline of corresponding ECG task $D(\text{HR})$.

test [17]. According to this test, a task set is schedulable if the total utilization of all tasks is less or equal to 1, where utilization is defined as the ratio between a task's worst-case execution time and its period. Any regular task τ_i has a constant utilization $U(\tau_i) = \frac{C_i}{T_i}$, but the utilization of E depends on the heart rate and varies for different instances: $U(E^i) = \frac{C^i(\text{HR})}{D^i(\text{HR})}$. Therefore, the task set would be schedulable if,

$$\sum_{\forall i} U(\tau_i) + \max_i U(E^i t) \leq 1$$
$$\sum_{\forall i} \frac{C_i}{T_i} + \max_i \frac{C^i(\text{HR})}{D^i(\text{HR})} \leq 1 \tag{2}$$

Note, although $\max_i U(E^i)$ find the maximum utilization of ECG task for any sampling instance, both $C^i(\text{HR})$ and $D^i(\text{HR})$ have only a limited number of options based on the HR threshold proposed in the paper. Therefore, $\max_i U(E^i)$ can be computed efficiently.

V. Design of CVD Detection Model

This section introduces the design of our CVD detection model, which combines ECG signal features and heart rate patterns to make accurate predictions. The model is built for real-time use, with a focus on both performance and efficiency.

A. Model Workflow

Figure 5 summarises the end-to-end signal flow. Throughout the diagram, α is the target sequence length after adaptive pooling, β is the number of heartbeats in the window, and γ is the number of diagnostic classes.

Input preparation. Each segment contains β consecutive R–R cycles, every cycle resampled to 256 samples, yielding an ECG tensor of length 256 β. A companion vector of the same cardinality (β) stores the inter-beat periods, capturing heart-rate variability.

ECG branch. A stem convolution with 8 β channels extracts low-level morphology; the signal is then processed by three Residual Blocks whose channel widths increase to 16 β, 32 β, and 64 β. Within each block, a squeeze-and-excitation (SE) unit re-weights channels according to their global importance. An adaptive average-pool then compresses the temporal dimension to α, producing a feature map of size $\alpha \times 64\ \beta$.

Period branch. The period vector passes through a fully-connected (FC) block, `Linear`→ `BatchNorm`→ `ELU`, expanding its representation to 64 β features. This transformation places temporal information in the same feature space as the pooled ECG map.

Global attention fusion. Flattened ECG features $\left(64\ \beta\alpha\right)$ are concatenated with the period embedding $\left(64\ \beta\right)$ and fed to a two-layer attention module. The resulting sigmoid mask modulates the ECG features, enabling the network to emphasise cycle regions whose relevance is corroborated by the instantaneous rhythm context.

Classification head. The attended ECG vector and the period embedding are re-concatenated and passed through an output block consisting of two FC layers; the final linear layer produces γ logits.

This architecture couples strong morphological feature learning (convolution + residual + SE) with rhythm-aware re-weighting (period branch + global attention) while maintaining a compact, throughput-oriented design suitable for real-time embedded inference.

B. Residual Blocks with Squeeze-and-Excitation Layers

A fundamental advancement in our model is the incorporation of SE layers within Residual Blocks, which enhances feature recalibration. This allows the network to emphasize informative features while suppressing less relevant ones. Mathematically, each Residual Block with an SE layer can be described as:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{\mathbf{W}_i\}) \otimes \mathbf{s} + \mathbf{x}, \tag{3}$$

where:

- $\mathbf{x}$ is the input feature map,
- $\mathcal{F}(\mathbf{x}, \{\mathbf{W}_i\})$ represents the residual function composed of convolutional operations,
- $\mathbf{W}_i$ denotes the learnable parameters,
- $\mathbf{s} \in \mathbb{R}^C$ is the scaling vector obtained through the SE layer.

The scaling vector $\mathbf{s} \in \mathbb{R}^C$ is computed as:

$$\mathbf{s} = \sigma\left(\mathbf{W}_2 \cdot \delta\left(\mathbf{W}_1 \cdot \text{AvgPool}(\mathbf{x})\right)\right), \tag{4}$$

where:

- $\text{AvgPool}(\mathbf{x})$ performs global average pooling,
- δ represents the ReLU activation function,
- σ is the sigmoid activation function,
- $\mathbf{W}_1$ and $\mathbf{W}_2$ are the weight matrices for the SE layers.

The element-wise multiplication $\otimes$ applies the recalibrated weights to the residual output, enhancing the network's ability to focus on salient features while mitigating the vanishing gradient problem inherent in deep architectures.

C. Global Attention Mechanism

Building upon the feature extraction capabilities, the Global Attention mechanism is a pivotal advancement that effectively integrates ECG features with period features to enhance the

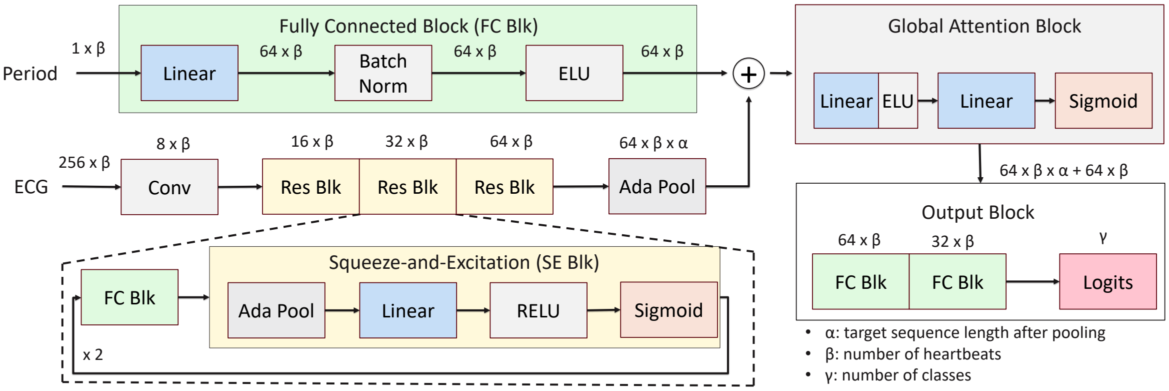


Fig. 5: The proposed model architecture illustrating the integration of Residual Blocks with SE layers, the Global Attention mechanism, and the Output Layer consisting of three fully connected layers for anomaly detection.

model's discriminative capability. This mechanism operates by computing attention weights that dynamically emphasize critical regions of the ECG signal based on both intrinsic ECG characteristics and temporal context provided by period features.

Formally, let $\mathbf{f}_{\text{ECG}} \in \mathbb{R}^{d_{\text{ECG}}}$ denote the flattened ECG feature vector and $\mathbf{f}_{\text{period}} \in \mathbb{R}^{d_{\text{period}}}$ represent the period feature vector. The combined feature vector is defined as:

$$\mathbf{f}_{\text{combined}} = \begin{bmatrix} \mathbf{f}_{\text{ECG}} \\ \mathbf{f}_{\text{period}} \end{bmatrix} \in \mathbb{R}^{d_{\text{ECG}}+d_{\text{period}}}. \tag{5}$$

The attention weights $\mathbf{a} \in \mathbb{R}^{d_{\text{ECG}}}$ are computed through a two-layer fully connected network:

$$\mathbf{h} = \text{ELU}(\mathbf{W}_1 \mathbf{f}_{\text{combined}} + \mathbf{b}_1), \tag{6}$$

$$\mathbf{a} = \sigma(\mathbf{W}_2 \mathbf{h} + \mathbf{b}_2), \tag{7}$$

where:

- $\mathbf{W}_1 \in \mathbb{R}^{m \times (d_{\text{ECG}}+d_{\text{period}})}$,
- $\mathbf{W}_2 \in \mathbb{R}^{d_{\text{ECG}} \times m}$,
- m is the dimensionality of the intermediate layer,
- $\mathbf{b}_1$ and $\mathbf{b}_2$ are the bias vectors,
- σ is the sigmoid activation function.

The computed attention weights are then applied to the ECG feature map $\mathbf{F}_{\text{ECG}} \in \mathbb{R}^{d_{\text{ECG}} \times T}$ (where T represents the temporal dimension):

$$\mathbf{F}_{\text{attended}} = \mathbf{a} \otimes \mathbf{F}_{\text{ECG}}, \tag{8}$$

where $\otimes$ denotes element-wise multiplication. This operation effectively highlights regions of the ECG signal that are most indicative of potential anomalies, guided by both the ECG data and the period context.

VI. AMS Framework and Anytime Model

This section describes how our run-time (1) *selects* an appropriate network depth on a beat-to-beat basis and (2) *realises* those depths inside a single parameter-shared CNN.

A. Adaptive Model Selection Framework

Shown in Algorithm 1, the AMS controller maintains a pool of three pre-trained networks—Lightweight, Moderate, and Advanced—whose inference latencies form a strict hierarchy. At every shifted-window arrival, it reads the instantaneous heart rate (HR), compares it with two fixed thresholds, and chooses the shallowest model that can meet the beat deadline.

- **Model Pool:** three CNN variants with increasing accuracy–latency trade-offs.
- **Selection Rule:** *High* HR (HR $\geq$ 90 bpm) → Lightweight; *Moderate* HR ($70 \leq$ HR < 90) → Moderate; *Low* HR (HR $<$ 70) → Advanced.

Algorithm 1: Adaptive Model Selection (AMS)

```
Input: ECG segment $E^i$, instantaneous HR HR
Output: Prediction $\hat{y}_i$
if HR $\geq 90$ then
    Model $\leftarrow$ LIGHTWEIGHT
else if $70 \leq$ HR $< 90$ then
    Model $\leftarrow$ MODERATE
else
    Model $\leftarrow$ ADVANCED
end
$\hat{y}_i \leftarrow$ Model($E^i$);
if InferenceTime(Model)> $D$(HR) then
    re-run on shallower model or flag timing anomaly
end
return $\hat{y}_i$
```

The two thresholds, 70 bpm and 90 bpm, correspond to the 33rd and 66th percentiles of the training distribution and can be re-tuned to trade precision for latency on other devices or patient cohorts. To prevent thrashing when the instantaneous HR drifts around a boundary, we smooth the measurement with a fixed-duration, peak-anchored window: after each window of length T_w closes, the next begins at the last R-peak observed and again spans T_w s, so isolated ectopic beats or missed detections have minimal impact on the averaged rate. In deployment, a microsecond-resolution watchdog can be enabled to pre-empt inference once elapsed time crosses a configurable fraction $kD(\text{HR})$ of the beat budget (e.g., $k = 0.8$); the segment is then re-evaluated on the next shallower model, and the strict latency hierarchy ensures the fallback still meets $D(\text{HR})$. Because AMS performs only two threshold checks and an infrequent watchdog test, its run-time overhead is negligible.

B. Anytime Model with Parameter Sharing

Storing three independent checkpoints would exhaust the memory budget of a smartwatch. Instead, we implement the three complexity levels as early-exit heads of a single *Anytime CNN*. A shared backbone of convolutional and residual–attention layers feeds:

- an **early exit** that returns a prediction after a global-average pool (high-HR windows),
- an **intermediate exit** that adds one residual+SE block (moderate HR),
- a **deep exit** that adds two further residual+SE blocks plus global attention (low HR).

Thus, shown in Algorithm 2, deeper predictions refine shallower ones without reloading weights; switching depth merely truncates the computation graph.

```
Algorithm 2: Anytime CNN Inference
  Input: ECG segment $E^i$, instantaneous HR HR
  Output: Prediction $\hat{y}_i$
  features ← Backbone($E^i$);
  if HR $\geq 90$ then
      $\hat{y}_i$ ← Lightweight_Head(features)
  else if $70 \leq$ HR $< 90$ then
      features ← Moderate_Block(features);
      $\hat{y}_i$ ← Moderate_Head(features)
  else
      features ← Advanced_Block(features);
      $\hat{y}_i$ ← Advanced_Head(features)
  end
  return $\hat{y}_i$
```

The network is trained with *deep supervision*; losses from the three exits are summed with equal weights so that each head remains independently accurate. At inference, AMS first chooses the required depth (Algorithm 1); the Anytime CNN then evaluates only up to that exit, saving both memory transfers and latency while maintaining a single compact checkpoint. This combination delivers the appropriate latency–accuracy point for every heart-rate window without incurring context-switch overhead.

VII. Experiments and Results

To evaluate our model's real-world effectiveness, we conducted a series of experiments focused on both classification performance and real-time feasibility. Using a large-scale ECG dataset and simulating deployment on embedded hardware, we tested how well our models detect cardiac anomalies under various heart rate conditions and resource constraints.

A. Dataset

We employ the PhysioNet 2021 Challenge Database[2] [25] as our primary data source. This comprehensive collection contains over 100,000 annotated ECG recordings from diverse clinical environments spanning four countries across three continents. Although the database includes multiple leads and covers 133 different disease categories, real-world wellness devices commonly capture only a single lead (often Lead I). Accordingly, we filtered the database to retain only those 72 disease classes that can reliably manifest in Lead I data.

Rather than classifying individual diseases directly, we assign each recording a *seriousness score* ranging from 0 to 4. Here, 0 indicates normal or minimally severe conditions, while 4 denotes the most critical or severe abnormalities. This condensed labeling scheme reflects practical realities in wellness devices: given unavoidable sensor limitations (e.g., single-lead availability, potential motion artifacts, and suboptimal signal quality), distinguishing among 72 specific diseases is less feasible in daily life scenarios. Instead, using a 0–4 severity scale allows the system to deliver timely alerts to the user and provide a general gauge of their heart health, aligning more closely with everyday monitoring use cases.

To ensure beat-level evaluation precision, we refined the dataset by selecting only the recordings with a single disease label, resulting in a dataset of 22,359 single-label ECG recordings ranging from 5 to 1800 seconds in length. By excluding multi-label recordings, we eliminate ambiguity in anomaly detection, enabling more accurate model training and evaluation at the individual beat level.

TABLE I: Class Distribution

Label	Training Set	Validation Set	Test Set
Label 0	145,866	36,467	45,583
Label 1	107,908	26,977	33,722
Label 2	42,760	10,690	13,363
Label 3	43,882	10,971	13,713
Label 4	10,106	2,526	3,158

Shown in Table I, we split the dataset into 80% training and 20% testing, and further split the training portion (80%) into 80% for actual training (64% of the original data) and 20% for validation (16% of the original data). This results in a final distribution of 64% for training, 16% for validation, and 20% for testing across all labels.

The overall label distribution percentages are consistent across these splits, maintaining the original class imbalance. Label 0 comprises approximately 33% of the entire dataset, while Label 1 makes up about 24.6%. Labels 2 and 3 account for around 9.8% and 10%, respectively. Label 4, the least represented class, makes up only about 2.3% of the dataset.

[2] Dataset: https://physionet.org/content/challenge-2021/1.0.3/#files

This imbalance mirrors real-world scenarios, where data for highly severe diseases is limited.

B. Experimental Setup

To train the proposed models, we utilize an NVIDIA GeForce RTX 3080 GPU for efficient batch processing and hyperparameter tuning. Once trained, we deploy these models on a Raspberry Pi 4 to evaluate real-time performance under constrained computational resources. The Raspberry Pi 4, which features a quad-core ARM Cortex-A72 processor, serves as an effective stand-in for commercial wearable devices (e.g., the Samsung Galaxy Watch 6 powered by dual-core ARM Cortex-A55 [24], [27]). This hardware similarity makes it a suitable testbed for verifying that the trained models adhere to stringent real-time requirements common in wearable health-monitoring scenarios.

We prepare the ECG data through a series of preprocessing steps to ensure consistency and improve signal quality. First, R-peaks are automatically detected using Hamilton's algorithm from the BioSPPY library [3], enabling segmentation of the continuous ECG waveforms into individual heartbeat cycles. Each cycle is then resampled to a fixed length of 256 data points to provide uniform input dimensions for the model. Finally, each preprocessed cycle is labeled according to its severity score (0–4), yielding a training-ready dataset aligned with our classification objective. These steps collectively ensure that the model receives clean, standardized ECG inputs, thereby facilitating more reliable model training and evaluation under embedded system constraints.

C. Experiments

We designed our experiments to evaluate two primary aspects: (1) the effectiveness of our proposed model architectures in accurately detecting cardiac anomalies under different temporal contexts, and (2) their computational efficiency in resource-constrained scenarios.

1) Model Performance Evaluation: We compare three model variants to investigate how architectural components impact diagnostic accuracy and runtime performance:

- Advanced Model: Combines Residual Blocks, SE layers, and a Global Attention mechanism for maximal feature extraction.
- Moderate Model: Omits the Global Attention mechanism while retaining Residual and SE layers to measure the attention mechanism's added value.
- Lightweight Model: Excludes both the attention mechanism and SE layers, significantly reducing the parameter count to favor faster inference.

Each model was trained independently using the same dataset and label structure. We used the Adam optimizer (learning rate 0.001, batch size 128) and early stopping based on validation loss. No sequential or fine-tuning strategy was applied across models to avoid bias. For the unified anytime model, all exit branches were jointly trained using multi-exit supervision, ensuring valid predictions at different depths for real-time adaptation.

To explore the influence of temporal context, each model is tested on 1, 2, 3, and 4 cardiac cycles. This variation allows us to see whether processing additional heartbeats yields substantive gains in accuracy or proves too costly in time. In addition, we benchmarked against several baselines:

1) A Deep Neural Network (DNN) architecture [19].
2) An Attention-based CNN-LSTM model [15], [29].
3) A CNN-Transformer Hybrid, integrating convolutional filters with Transformer layers [4], [20].
4) A Temporal Convolutional Network (TCN) [13].

In each case, we measure standard classification metrics (e.g., accuracy, F1-score) alongside runtime indicators (e.g., per-inference latency) on the embedded platform. This approach captures both the quality of the predictions and the practicality of deploying these architectures in real-world, multitasking wearable environments.

2) Real-Time System Implementation: To assess the responsiveness of each model variant across varying heart rate conditions, we constructed a real-time ECG processing pipeline. In this pipeline, the device continuously monitors incoming ECG data along with the current HR, attempting to complete each inference within predefined time thresholds. By comparing actual inference durations against these deadlines, we can evaluate the real-time viability of the models under different heart rate scenarios.

Table II outlines the dataset size and processing time thresholds corresponding to distinct heart rate conditions. Although initially informed by typical physiological observations, these thresholds are defined to reflect maximum allowable latencies necessary for preserving real-time responsiveness. For instance, the threshold of 1.5 ms under the *High* HR scenario indicates that any inference taking longer could impede critical real-time functions (such as user-interface updates or sensor fusion processes), compromising system performance.

TABLE II: Settings for Different Heart Rate Conditions

Setting	**HR (beats/minute)**	**Sample Size**	**Threshold (ms)**
High	$\text{HR} \geq 90$	348,227	1.5
Mid	$70 \leq \text{HR} < 90$	336,238	1.75
Low	$\text{HR} < 70$	336,051	2

Practically, these thresholds may be adjusted based on specific application contexts or tighter performance criteria. For example, during periods of increased physiological stress or simultaneous critical tasks, real-time thresholds might be tightened to prioritize rapid response. Conversely, under lower heart rate conditions or reduced concurrent task demands, the thresholds can be relaxed, allowing for more comprehensive ECG anomaly analyses. This framework enables thorough evaluation of each model's scalability regarding inference speed and accuracy across varying heart rate conditions.

3) Evaluation Metrics: The performance of the disease detection models was evaluated using the following metrics:

- Cycles: Number of ECG cycles used as input.
- Accuracy (ACC): Proportion of correctly classified instances.

- Precision (PREC): Fraction of true positives among predicted positives.
- Recall (REC): Proportion of true positives correctly identified among all actual positives.
- F1-Score (F1): Harmonic mean of precision and recall, providing a balanced measure of model performance.
- Model Size (MB): Storage requirements, indicating feasibility for deployment on resource-constrained devices.
- Inference Time (ms): Average execution time per inference on a Raspberry Pi 4.
- Deadline Misses (DDL Miss): Number of missed deadlines out of 1,000 data samples.

D. Experimental Results

Table III presents the performance metrics of various models across different numbers of ECG cycles. In addition, Figure 6 shows the detailed performance for our proposed AMS model. The key findings from the results are as follows:

The Advanced model achieved the highest accuracy of 92.6% with two ECG cycles, indicating its strong ability to learn from additional data. However, its increased model size and inference time at higher cycle counts may limit its suitability for real-time applications on resource-constrained devices. The Lightweight model demonstrated commendable performance with minimal computational requirements, achieving 86.5% accuracy at two cycles. Its small model size and fastest inference times make it well-suited for embedded systems with limited resources. The AMS model offered the best balance between accuracy and efficiency, reaching 91.5% accuracy with two cycles while maintaining low inference times and zero deadline misses, making it ideal for real-time ECG analysis where both promptness and precision are critical.

In contrast, the DNN model exhibited low accuracy despite its computational efficiency, limiting its reliability for practical ECG classification tasks. The CNN-LSTM model, although delivering higher accuracy, suffered from prolonged inference times and frequent deadline misses, rendering it unsuitable for real-time applications. Similarly, the Transformer-based model demonstrated suboptimal accuracy combined with significant computational overhead, making it impractical for deployment in resource-constrained settings. Finally, the TCN model, while showing better performance among the baselines, still lagged behind our proposed models in terms of both accuracy and real-time responsiveness.

Overall, the results suggest that the AMS and Lightweight models are the most promising for real-time cardiovascular disease detection in embedded systems, offering an optimal trade-off between accuracy and computational efficiency.

VIII. Discussion

In this section, we reflect on the model's performance trends, explore its practical deployment in real-time wearable systems, and outline key limitations and future directions.

A. Performance Insights

Table III clarifies three intertwining trends—*context benefit*, *computational cost*, and *deadline pressure*—that jointly shape real-time feasibility. First, adding a second cardiac cycle markedly improves every architecture (e.g., Advanced rises from 87.3% to 92.6% accuracy and Lightweight from 82.4% to 86.5%), confirming that one extra beat supplies enough temporal context for the network to stabilise morphological variability. A third and fourth cycle, however, yield diminishing or negative returns: accuracy falls back to 88.1% and 86.5% for Advanced while latency more than quadruples (1.02 ms → 8.49 ms). This plateau indicates that longer segments start to blur transient abnormalities and simultaneously push the model into the non-real-time regime.

Second, inference time grows almost linearly with input length (slope $\approx$ 1.4–1.9 ms per additional cycle for Advanced), whereas memory footprint grows super-linearly because deeper layers activate more channels. For instance, Advanced at four cycles occupies 44.6 MB—over 200× the Lightweight model at one cycle—exceeding the cache budget of most embedded SoCs. These costs translate directly into deadline misses: once latency exceeds the 1.5–2 ms envelopes in Table II, 100% of the inferences over-run.

Third, the AMS+Anytime configuration exploits both observations. By supervising all exit heads jointly, it inherits the two-cycle accuracy sweet-spot (91.5%) while shaving 31% latency and 34% memory compared with the stand-alone Advanced model. Crucially, zero deadline misses are retained across *all* heart-rate regimes because AMS dispatches the early-exit path when the instantaneous HR shortens the deadline. In high-HR windows, the framework therefore behaves like the Lightweight network (0.57 ms), whereas in low-HR windows it escalates to the deeper path without incurring a context switch.

The baselines reinforce these dynamics. CNN-LSTM achieves 87.3% accuracy at two cycles yet needs 3.33 ms and misses every deadline; Transformers are both slower and less accurate; TCNs narrowly trail the proposed models but still violate timing once three cycles are processed. Overall, the results demonstrate that (i) two-cycle context is optimal on our dataset, (ii) architectural refinement—SE layers plus global attention—drives accuracy gains, and (iii) adaptive depth selection is indispensable for meeting millisecond-level timing constraints on embedded platforms.

B. Practical Deployment and Clinical Impact

Resource-constrained wearables must juggle power, thermal, and multitasking limits while still delivering reliable cardiac insights. Our framework satisfies these competing demands by dynamically scaling model depth: under heavy system load or elevated heart-rate deadlines, it executes the lightweight path, preserving headroom for user-interface, radio, and sensor-fusion tasks; in calmer periods, it escalates to the moderate or advanced path for finer-grained analysis. Because all paths share a common backbone in the *Anytime Model*, switching incurs no weight reload and only marginal SRAM overhead, keeping the total footprint below 5 MB in the two-cycle AMS configuration. This combination—adaptive model selection plus parameter sharing—enables continuous,

TABLE III: Performance Metrics of Model Variants Across Different ECG Cycles

Model	Cycles	ACC (%)	PREC (%)	REC (%)	F1 (%)	Size (MB)	Inference (ms)	DDL Miss
DNN [19]	1	55.7	56.1	55.4	54.8	0.11	0.05	0
	2	60.9	60.7	60.8	60.6	0.29	0.12	0
	3	58.6	57.2	56.1	57.2	0.58	1.30	13
	4	57.4	56.3	56.0	58.8	1.42	3.41	1000
CNN-LSTM [15], [29]	1	81.1	80.2	81.1	81.3	3.25	2.67	1000
	2	87.3	87.7	87.2	87.6	5.56	3.33	1000
	3	83.7	83.2	83.5	84.1	12.18	5.29	1000
	4	80.4	79.5	80.0	79.8	27.51	16.13	1000
Transformer [4], [20]	1	65.2	64.8	65.0	64.9	10.52	4.31	1000
	2	70.1	69.8	69.5	69.7	18.73	7.62	1000
	3	67.8	67.2	67.5	67.3	32.84	13.91	1000
	4	65.9	65.5	65.2	65.4	58.21	25.43	1000
TCN [13]	1	84.5	84.2	84.6	84.4	5.21	1.34	52
	2	86.1	86.0	86.2	86.1	7.93	2.18	979
	3	85.4	85.1	85.3	85.2	14.29	4.76	1000
	4	84.8	84.5	84.7	84.6	25.63	9.32	1000
Advanced	1	87.3	87.1	87.1	87.0	2.1	1.02	0
	2	**92.6**	**91.8**	**92.5**	**91.1**	6.22	1.94	431
	3	88.1	87.7	88.4	88.0	19.5	4.30	1000
	4	86.5	86.0	86.8	86.4	44.6	8.49	1000
Moderate	1	85.1	85.7	85.4	85.0	0.49	0.68	0
	2	87.8	88.5	88.0	87.7	2.09	1.79	259
	3	86.5	86.2	86.8	86.5	5.7	3.94	1000
	4	84.0	84.7	84.3	84.7	12.8	7.31	1000
Lightweight	1	82.4	82.8	83.5	83.1	**0.18**	**0.39**	0
	2	86.5	87.2	86.0	86.6	0.44	1.05	0
	3	84.7	83.9	84.5	83.2	1.8	2.62	1000
	4	82.8	82.5	82.0	82.8	4.3	5.22	1000
AMS+Anytime	1	86.6	86.2	86.4	86.8	1.1	0.57	0
	2	91.5	90.7	91.0	90.6	4.12	1.33	0
	3	87.2	87.9	86.5	87.2	11.6	3.13	1000
	4	85.4	85.5	85.0	85.8	33.9	6.96	1000

on-wrist inference without violating the millisecond-level budgets listed in Table II.

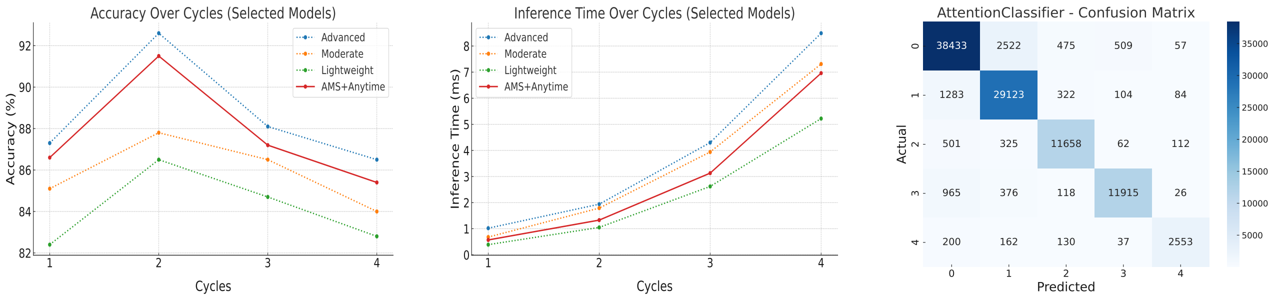
Clinically, such adaptability broadens the utility of single-lead devices across daily activities. During rest or sleep, longer R–R intervals give the system time to run the advanced branch, boosting sensitivity to subtle bradyarrhythmias; during vigorous exercise, it seamlessly falls back to the lightweight branch, maintaining real-time alerts despite faster rhythms and higher motion noise. Providers can therefore prescribe the same wearable for post-event monitoring, rehabilitation, or preventive fitness, confident that severity-graded alarms remain consistent. For consumers, a single device now covers sleep tracking, workout feedback, and routine health checks without manual mode switching, encouraging sustained engagement. Remaining deployment challenges—user education, alarm-fatigue mitigation, and EHR integration—can be addressed at the application layer, building on the robust, adaptive inference core demonstrated here.

C. Limitations and Future Directions

Despite its considerable promise, our framework faces several constraints that merit further exploration. First, while our dataset and experiments introduced controlled noise to mimic real-world conditions, practical deployments may experience more severe artifacts caused by sensor misplacement, motion disturbances, or environmental interference. Addressing these issues might require advanced denoising algorithms or domain adaptation techniques that enhance model robustness without overburdening the embedded hardware.

Another limitation concerns the reliance on labeled training data. Large, meticulously annotated ECG datasets can be difficult to assemble, particularly for rare cardiac conditions. Semi-supervised or self-supervised learning paradigms could alleviate this data bottleneck by harnessing unlabeled physiological signals to learn robust feature representations. Moreover, although the Raspberry Pi 4 offers a reasonable analog to commercial wearable platforms, a wide range of hardware and firmware variations exist across manufacturers. Testing our framework on these different platforms would confirm its broader applicability and uncover potential optimizations tied to specific architectures or system constraints.

An additional avenue for improvement lies in refining the classification scheme itself. Currently, we assign uniform importance to all severity levels of cardiac anomalies during training. However, in real-world scenarios, it may be far more critical to avoid misclassifying severe abnormalities as normal than to occasionally misclassify a normal reading as abnormal.



(a) Accuracy Trend Visualization

(b) Inference Trend Visualization

(c) Confusion Matrix

Fig. 6: Visualization of Accuracy and Inference Trends, and Confusion Matrix for Advanced Model with Two Cycles.

Future work could incorporate asymmetric cost functions or class-weighting strategies to prioritize higher-severity conditions, ensuring that rare yet critical heart anomalies are accurately captured.

IX. Related Work

Single-Lead ECG Processing in Wearable Devices: Single-lead ECG devices have gained prominence due to their compactness and ease of integration into wearable systems. Studies have demonstrated that single-lead ECG can reliably detect various cardiac abnormalities, including arrhythmias [8]. The simplicity of single-lead systems makes them ideal for continuous monitoring in consumer-grade wearables [26]. However, the potential decrease in diagnostic accuracy necessitates advanced signal processing and machine learning techniques to compensate for the limited data [31].

Real-Time ECG Anomaly Detection: Real-time anomaly detection in ECG signals is critical for timely intervention. Traditional methods rely on threshold-based algorithms, which may not generalize well across diverse populations and noisy environments [7]. Recent advances leverage machine learning, particularly deep learning models, to enhance accuracy and robustness [23]. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been employed to automatically extract complex features from raw ECG data, enabling precise identification of irregular heartbeats [1].

Dynamic Model Scaling for Resource-Constrained Devices: Dynamic model scaling adjusts the complexity of machine learning models based on processing requirements and available computational resources, beneficial for wearable devices under power and performance constraints [2]. Techniques like model pruning and quantization enable real-time performance without compromising accuracy [11]. By dynamically selecting architectures that balance speed and accuracy, wearable ECG systems maintain continuous monitoring capabilities under varying conditions.

Deep Learning Architectures for ECG Anomaly Detection: Deep learning architectures have revolutionized ECG anomaly detection by providing end-to-end solutions that learn hierarchical features from raw signals. CNNs capture local temporal patterns, while RNNs and Long Short-Term Memory (LSTM) networks model sequential dependencies [12]. Hybrid architectures combining CNNs with attention mechanisms enhance feature extraction and focus on critical ECG segments [14]. These advanced models achieve high accuracy in classifying cardiac conditions like atrial fibrillation and ventricular tachycardia [28].

Integration of Adaptive Techniques in ECG Monitoring Systems: The integration of adaptive techniques such as shifted window approaches and dynamic model scaling within ECG monitoring systems represents a novel advancement. While studies have explored these techniques individually, their combined application for real-time, single-lead ECG analysis remains under-explored [9]. Our proposed system builds upon these foundations by implementing an adaptive shifted window mechanism coupled with dynamic model selection based on heart rate variability. This integration ensures efficient resource utilization and maintains high diagnostic performance, addressing critical challenges in real-time ECG monitoring on wearable devices.

X. Conclusion and Future Work

In this work, we presented an anytime, heart-rate–aware ECG-analysis framework that unifies lightweight, moderate, and advanced exits inside a single parameter-shared CNN and schedules the shallowest path that can satisfy each beat's timing budget, achieving high diagnostic accuracy while respecting real-time constraints on an embedded prototype. Going forward we will (i) reinterpret those timing budgets in collaboration with clinicians to align latency targets with true intervention windows, (ii) validate the system on wrist-worn prototypes using live, noise-prone ECG streams rather than an offline PhysioNet dataset, (iii) profile energy on an MCU-class SoC to confirm battery feasibility, (iv) replace the fixed 70/90 bpm thresholds with personalised or learned switching logic, (v) harden the pipeline against motion artefacts via denoising and adversarial augmentation, (vi) restore diagnostic granularity by tackling class imbalance with hierarchical or focal training instead of collapsing 72 labels into five, and (vii) extend the schedulability analysis to include context-switch and interrupt overheads so that deadline guarantees hold on a wearable device with real-time operating system.

Acknowledgement

This work was supported in part by the National Science Foundation under Grant CMMI 2246672.

References

[1] Z. I. Attia, P. A. Friedman, P. A. Noseworthy, et al. An artificial intelligence-enabled ECG algorithm for the identification of patients with atrial fibrillation during sinus rhythm: A retrospective analysis of outcome prediction. *The Lancet*, 394(10201):861–867, 2019.

[2] H. Cai, C. Gan, and S. Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.

[3] C. Carreiras, A. P. Alves, A. Lourenço, F. Canento, H. Silva, A. Fred, et al. BioSPPy: Biosignal processing in Python, 2015–. [Online; accessed ¡today¿].

[4] R. Cheng, Z. Zhuang, S. Zhuang, L. Xie, and J. Guo. MSW-Transformer: Multi-scale shifted windows transformer networks for 12-lead ECG classification. *arXiv preprint arXiv:2306.12098*, 2023.

[5] S. S. Chugh, K. Reinier, C. Teodorescu, A. Evanado, E. Kehr, M. Al Samara, R. Mariani, K. Gunson, and J. Jui. Epidemiology of sudden cardiac death: clinical and research implications. *Progress in Cardiovascular Diseases*, 51(3):213–228, Nov-Dec 2008. PMID: 19026856; PMCID: PMC2621010.

[6] G. D. Clifford, F. Azuaje, and P. E. McSharry. *Advanced Methods and Tools for ECG Data Analysis*. Artech House, 2006.

[7] M. Elgendi. Fast qrs detection with an optimized knowledge-based method: Evaluation on 11 standard ecg databases. *PLOS ONE*, 8(9):e73557, 2013.

[8] A. Galli, F. Ambrosini, and F. Lombardi. Holter monitoring and loop recorders: From research to clinical practice. *Arrhythmia & Electrophysiology Review*, 5(2):136–141, 2016.

[9] U. Gupta, N. Paluru, D. Nankani, K. Kulkarni, and N. Awasthi. A comprehensive review on efficient artificial intelligence models for classification of abnormal cardiac rhythms using electrocardiograms. *Heliyon*, 10(5):e26787, 2024.

[10] P. Hamilton. Open source ecg analysis. In *Computers in Cardiology*, pages 101–104, 2002.

[11] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *4th International Conference on Learning Representations (ICLR)*, 2016.

[12] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature Medicine*, 25(1):65–69, 2019.

[13] T. M. Ingolfsson, X. Wang, M. Hersche, A. Burrello, L. Cavigelli, and L. Benini. Ecg-tcn: Wearable cardiac arrhythmia detection with a temporal convolutional network. *arXiv preprint arXiv:2103.13740*, 2021.

[14] M. R. Islam, M. Qaraqe, K. Qaraqe, and E. Serpedin. Cat-net: Convolution, attention, and transformer based network for single-lead ecg arrhythmia classification. *Biomedical Signal Processing and Control*, 93:106211, 2024.

[15] Y. Li, N. Sui, A. Gehi, C. Guo, and Z. Guo. Cardiacrt-nn: Real-time detection of cardiovascular disease using self-attention cnn-lstm for embedded systems. In X. Le and Z. Zhang, editors, *Advances in Neural Networks – ISNN 2024*, pages 610–621, Singapore, 2024. Springer Nature Singapore.

[16] Y. Li, N. Sui, C. Xu, A. Gehi, and Z. Guo. Poster abstract: Real-time cardiovascular disease detection via abnormal electrocardiogram cycles on embedded systems. In *2024 23rd ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 287–288, 2024.

[17] C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)*, 20(1):46–61, 1973.

[18] M. S. Maron, E. J. Rowin, and B. J. Maron. Letter by maron et al regarding article, ”sudden cardiac death in national collegiate athletic association athletes”. *Circulation*, 150(12):e255–e256, sep 2024. Epub 2024 Sep 16; PMID: 39283935.

[19] A. Nainwal, Y. Kumar, and B. Jha. An ecg classification using dnn classifier with modified pigeon inspired optimizer. *Multimedia tools and applications*, 81(7):9131–9150, 2022.

[20] A. Natarajan, Y. Chang, S. Mariani, A. Rahman, G. Boverman, S. Vij, and J. Rubin. A wide and deep transformer neural network for 12-lead ECG classification. In *2020 Computing in Cardiology*, pages 1–4. IEEE, 2020.

[21] W. H. Organization. Cardiovascular diseases (cvds). *WHO*, 2021. https://www.who.int/health-topics/cardiovascular-diseases.

[22] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers. A review of wearable sensors and systems with application in rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 9(1):21, 2012.

[23] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng. Cardiologist-level arrhythmia detection with convolutional neural networks. *arXiv preprint arXiv:1707.01836*, 2017.

[24] Raspberry Pi Foundation. Raspberry pi 4 model b. https://www.raspberrypi.org/products/raspberry-pi-4-model-b/. Accessed: 2025-03-14.

[25] M. Reyna, N. Sadr, E. Perez Alday, A. Gu, A. Shah, C. Robichaux, A. Rad, A. Elola, S. Seyedi, S. Ansari, et al. Will two do? varying dimensions in electrocardiography: the physionet/computing in cardiology challenge 2021. In *2021 Computing in Cardiology (CinC)*, pages 1–4. IEEE, 2021.

[26] A. L. P. Ribeiro, A. L. Ribeiro, et al. Tele-electrocardiography and big data: The code (r) evolution. *Arquivos Brasileiros de Cardiologia*, 115(5):774–776, 2020.

[27] Samsung Electronics. Samsung galaxy watch 6. https://www.samsung.com/global/galaxy/galaxy-watch6/. Accessed: 2025-03-14.

[28] N. Strodthoff, P. Wagner, T. Schaeffter, and W. Samek. Deep learning for ECG analysis: Benchmarks and insights from ptb-xl. *IEEE Journal of Biomedical and Health Informatics*, 25(5):1519–1528, 2021.

[29] Z. Wang, X. Zhou, X. Liu, and H. Zhang. Automatic cardiac arrhythmias classification using cnn and attention-based rnn network. *Healthcare Technology Letters*, 10(3):67–74, 2023.

[30] J. Xu, Y. Zhang, M. Xie, W. Wang, and D. Zhu. Real-time intelligent on-device monitoring of heart rate variability with ppg sensors. *Journal of Systems Architecture*, 154:103240, 2024.

[31] Y. Xu, M. Luo, T. Li, and G. Song. Ecg signal de-noising and baseline wander correction based on ceemdan and wavelet threshold. *Sensors*, 17(12):2754, 2017.