Efficient ECG Arrhythmia Detection on FPGA using Machine Learning and Fiducial Windowing

K. P. Nandini
School of ECE, REVA University, India
r20pec20@ece.reva.edu.in (corresponding author)

G. Seshikala
School of ECE, REVA University, India
seshikala.g@reva.edu.in

Received: 12 November 2024 | Revised: 1 December 2024 and 13 December 2024 | Accepted: 18 December 2024

 DOI: https://doi.org/10.48084/etasr.9589

ABSTRACT

This work presents an efficient FPGA-based system for real-time detection of ECG arrhythmias using machine learning and fiducial windowing techniques. The proposed system integrates FPGA hardware acceleration to achieve low latency and high energy efficiency while maintaining superior classification accuracy, making it well-suited for portable health monitoring devices. ECG signals are preprocessed with a Butterworth filter to remove noise, followed by feature extraction through Discrete Wavelet Transform (DWT). The fiducial windowing method identifies key ECG components such as the P-wave, the QRS complex, and the T-wave, allowing the extraction of clinically relevant features. These features are then classified using a machine learning model implemented on an FPGA, allowing for rapid and accurate arrhythmia detection. The hardware-based solution significantly outperforms traditional software implementations in terms of real-time performance and power consumption. The proposed system achieved an impressive accuracy of 99.7%, a processing speed of 0.723 s, and a power consumption of 0.42 mW. The design was implemented using Xilinx Vivado 2022 EDA tools on the Xilinx PYNQ FPGA platform. This study demonstrates the potential of FPGA-based machine learning systems for efficient and reliable real-time ECG analysis, paving the way for advanced wearable health monitoring applications.

Keywords-ECG; FPGA; DWT; fiducial windowing; AI/ML; CNN; PYNQ Z2

I. INTRODUCTION

Electrocardiography (ECG) plays a vital role in monitoring heart health, particularly in detecting arrhythmias, which are irregular heart rhythms caused by abnormal electrical signals. Common arrhythmias include atrial fibrillation, ventricular tachycardia, and sudden cardiac arrest. Accurate detection of these irregularities is crucial to managing cardiovascular disease, one of the leading causes of death worldwide. Traditional ECG systems are based on manual analysis, which can be time-consuming and prone to errors. This limitation highlights the need for intelligent and automated systems capable of accurately identifying arrhythmias. Machine Learning (ML) has become increasingly popular in ECG analysis due to its ability to make precise predictions from data. However, the deployment of ML models in real time, especially in portable devices, poses challenges in terms of computational demands, power usage, and response time.

Field-Programmable Gate Arrays (FPGAs) offer an effective solution for real-time ECG monitoring in portable and wearable devices by combining the speed of hardware-level processing with the flexibility of software. FPGAs allow high-speed, parallel processing with minimal energy consumption, which is essential for real-time ECG signal analysis. ML algorithms such as Support Vector Machines (SVMs), decision trees, and Convolutional Neural Networks (CNNs) have shown effectiveness in arrhythmia detection but require optimization for deployment on resource-constrained devices. Techniques such as model pruning and quantization reduce computational demands while maintaining model accuracy, making them suitable for FPGA implementation. Feature extraction is crucial for efficient ECG analysis, with components such as the P-wave, the QRS complex, and the T-wave aiding in arrhythmia detection. Fiducial windowing focuses on these segments, simplifying data and enhancing processing efficiency. In FPGA-based ECG systems, fiducial windowing during preprocessing reduces noise and data volume, optimizing performance and resource efficiency for ML models.

The proposed FPGA-based system combines ML with fiducial windowing for efficient real-time arrhythmia detection. By extracting features in real time, the system classifies signals as normal or arrhythmic, leveraging a pre-trained ML model optimized through pruning and quantization. Implemented in a pipeline structure, each FPGA stage processes a part of the ECG signal analysis, ensuring low-latency and low-power performance ideal for portable devices. This approach provides a robust solution to continuous real-time ECG monitoring, balancing high classification accuracy with resource efficiency [1]. Real-time ECG detection using FPGA can ensure low latency and secure data processing, optimizing the computational efficiency of embedded systems [2, 3]. The objectives and major contributions of this work include:

- Develops a real-time FPGA-based system for efficient and accurate ECG arrhythmia detection.
- Integrates ML models with fiducial windowing for precise classification of key ECG features.
- Employs advanced signal processing techniques such as Butterworth filtering and DWT for noise reduction and feature optimization.
- Ensures low latency, energy efficiency, and robust performance to enable portable and wearable health monitoring applications.

The field of ECG arrhythmia detection has seen significant advances through various methods, including traditional signal processing techniques, machine learning, and hardware implementations such as FPGAs. Historically, ECG analysis was performed by manual interpretation, which is prone to variability and errors. Early automated systems employed rule-based algorithms, utilizing time-domain and frequency-domain analyses. Techniques such as Fourier and wavelet transforms have been utilized to extract features from ECG signals. The incorporation of ML techniques has revolutionized ECG analysis by automating the detection process and improving accuracy. Researchers have used various ML algorithms, including SVMs, decision trees, and neural networks, to classify arrhythmias effectively. More recently, deep learning approaches, particularly CNNs, have gained traction due to their ability to learn hierarchical feature representations directly from raw ECG data.

Windowing techniques, including fiducial windowing, have emerged as effective methods for feature extraction in ECG signals, focusing on key points within the ECG waveform [4]. This approach allows efficient extraction of relevant features while reducing computational complexity. Previous studies have shown that using windowing techniques significantly improved the classification accuracy of arrhythmias, as it allows the model to focus on critical segments of the ECG waveform. FPGAs have become a popular tool in ECG monitoring systems due to their low latency and real-time processing capabilities. Researchers have developed hybrid architectures that combine hardware and software processing to optimize arrhythmia detection [1].

The integration of AI and ML in ECG analysis has improved accuracy and reliability. AI-driven frameworks use deep learning techniques for automatic feature extraction and classification of ECG signals. Transfer learning has also been explored to address limited labeled datasets [1]. The ECG arrhythmia detection landscape is rapidly evolving, driven by advances in ML, windowing techniques, and FPGA implementations. Challenges, such as data variability and interpretability, remain in ML models [2]. In [4], ECG bio-identification was performed using a hybrid of fiducial and non-fiducial techniques. This study introduced a fiducial windowing method that segments the ECG signal around the QRS complex, combined with Short-Time Fourier Transform (STFT) and histogram-based analysis for feature extraction. This method enhances identification accuracy by leveraging both temporal and spectral features, demonstrating robustness against noise and variability in ECG signals [4].

II. DESIGN OF THE PROPOSED SYSTEM

Figure 1 shows the block diagram of the proposed system, which consists of several stages for ECG signal processing, starting with signal acquisition, followed by preprocessing, feature extraction, and ML-based classification. Initially, the ECG signal is filtered with a Butterworth filter to eliminate noise and artifacts. It is then processed using a 2-level 1D Discrete Wavelet Transform (DWT) to extract relevant frequency components. Fiducial windowing is employed to identify key features in the signal [1]. The use of fiducial windowing improves feature extraction accuracy, leading to more reliable arrhythmia classification. These features are input into a machine learning classifier deployed on PYNQ FPGA allowing for comparative analysis of performance, efficiency, and accuracy [2].

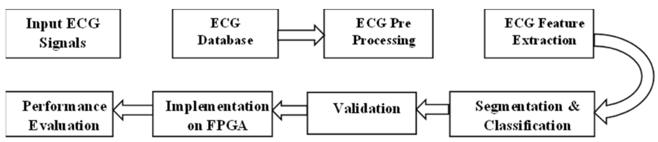

Fig. 1. Overall block diagram.

III. METHODOLOGY

The MIT-BIH Arrhythmia Database, a key ECG signal resource accessible through PhysioNet, contains 48 half-hour recordings from 47 subjects. Data is stored in .dat files for signals and header files for metadata. With the wfdb.rdrecord() function, ECG signals and metadata were loaded into a 2D array format, allowing detailed analysis and feature extraction, which is essential for accurate classification and diagnosis. Each recording in the MIT-BIH database spans 30 minutes and is comprehensively annotated with detailed labels [5]. Visualizing raw ECG signals using matplotlib helps reveal signal characteristics, such as the QRS complex, and assess quality by identifying issues such as baseline drift and noise. Key ECG components include the P-wave (atrial depolarization), QRS Complex, T-wave, and RR interval, aiding heart rate and arrhythmia analysis.

The transfer function $H(s)$ for an n^{th} order Butterworth filter can be mathematically described as:

$$H(s) = \frac{1}{1+\left(\frac{s}{\omega_c}\right)^{2n}} \tag{1}$$

The Butterworth filter used for ECG preprocessing employs two cutoff frequencies: a low cutoff frequency (ω_{low}) at 0.5 Hz to eliminate baseline noise and a high cutoff frequency (ω_{high}) at 40 Hz to suppress high-frequency noise. These filters are essential for normalizing frequencies in digital implementations, utilizing (2) and (3).

$$\omega'_{low} = \frac{2 * \omega_{low}}{f_s} \quad (2)$$

$$\omega'_{high} = \frac{2 * \omega_{high}}{f_s} \quad (3)$$

The Butterworth filter is implemented using the bilinear transformation method, converting the analog filter design into a digital one as:

$$s = \frac{2f_s}{1-z^{-1}} \quad (4)$$

The resulting digital filter can then be expressed in the form of a difference:

$$y[n] = b_0x[n] + b_1x[n-1] + \ldots + b_Nx[n-N] - a_1y[-1] - \ldots - a_My[-M] \quad (5)$$

This content explains the filtering process for ECG signals. The output signal $y[n]$ represents the filtered ECG, while the input $x[n]$ is the raw noisy ECG [6]. Figure 2 shows a filtered ECG signal, removing noise and artifacts for a cleaner waveform. The plot shows vertical peaks corresponding to QRS complexes, crucial for heart condition diagnosis. Figure 3 shows the filtered ECG signal visualization for the first 100 samples.

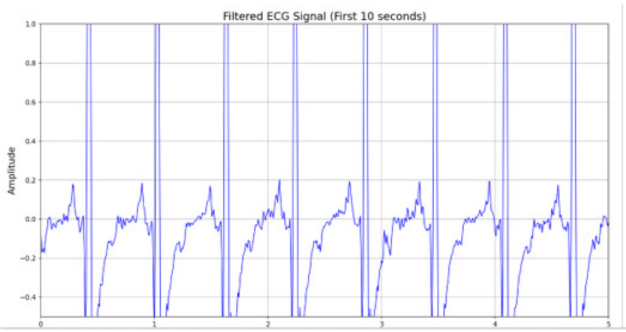

Fig. 2. Filtered ECG signal.

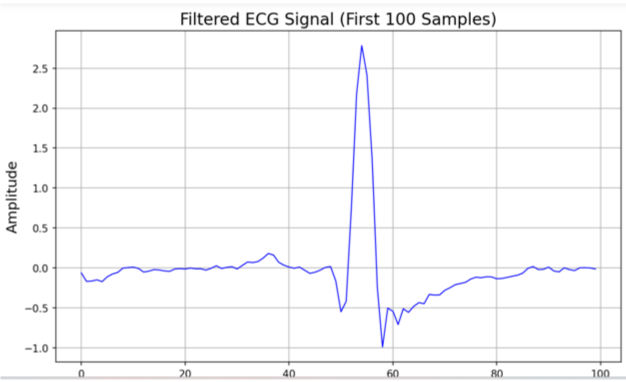

Fig. 3. Filtered ECG signal visualization (first 100 samples).

A. Feature Extraction

The ECG signal is processed using a 2-level 1D DWT for segmentation and analysis, separating key components such as the R-wave and the T-wave [4, 7]. Fiducial windowing locates the R-wave peak and T-wave onset, extracting the R-R interval, wave amplitude, and slope for diagnostic analysis.

$$X_{j,k} = \Sigma x[n] \cdot \psi_{j,k}[n] \quad (6)$$

where $X_{j,k}$ represents the wavelet coefficient at scale j and position k, $x[n]$ is the ECG signal, and $\psi_{j,k}[n]$ is the wavelet function at scale j and translation k. In the 2-level DWT, the ECG signal is decomposed into approximation (A) and detail (D) coefficients. At the first level, the signal is divided into:

$$x(t) = A_1 + D_1 \quad (7)$$

At the second level, the approximation A_1 is further decomposed into:

$$A_1 = A_2 + D_2 \quad (8)$$

So the overall signal representation becomes:

$$x(t) = A_2 + D_2 + D_1 \quad (9)$$

where A_2 denotes the low-frequency components corresponding to the baseline of the ECG, D_2denotes the higher frequency components corresponding to features such as the QRS complex, and D_1 denotes the highest frequency components, including noise and fine details [8]. Figure 4 shows a filtered signal plot with an amplitude and sample number over 10,000 and Figure 5 illustrates the frequency response of a bandpass filter.

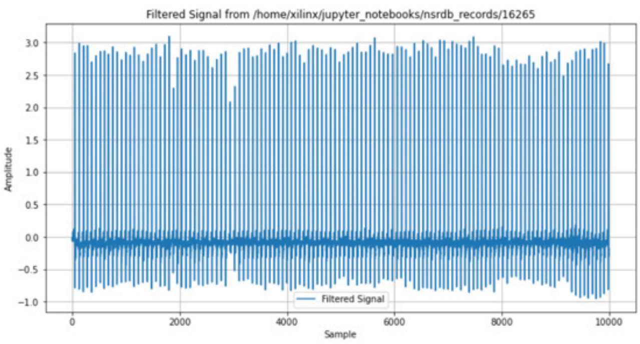

Fig. 4. Filtered signal plot with amplitude.

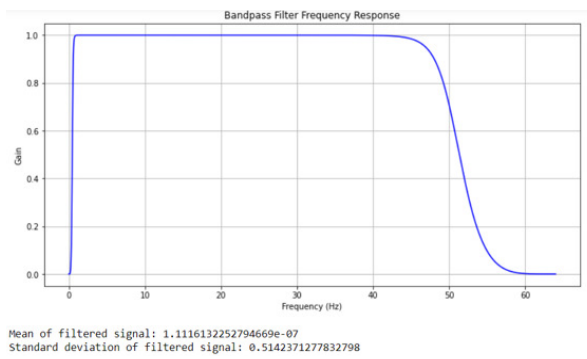

Fig. 5. Frequency response.

B. Fiducial Windowing for Key Feature Extraction from Fiducial Points

Fiducial windowing is a technique that isolates key points in an ECG signal for focused analysis of arrhythmias [4]. Once identifying fiducial points such as the R-wave peak and the T-wave onset, key features are extracted from the ECG signal to assess heart activity and diagnose cardiac conditions. The R-R interval, the time difference between successive R-wave peaks, is a critical feature that indicates heart rate and its variability and can be calculated using:

$$RR_i = t_{R_{i+1}} - t_{R_i} \tag{10}$$

where t_{R_i} and $t_{R_{i+1}}$ represent the times of consecutive R-wave peaks.

Wave amplitude, including that of the R-wave, the P-wave, and the T-wave, is measured as the difference in voltage between the peak and baseline. For example, the R-wave amplitude is given by:

$$A_R = x(t_R) - x_{baseline} \tag{11}$$

where $x(t_R)$ is the signal value at the R-wave peak, and $x_{baseline}$ is the signal value at the baseline.

The slope of the ECG waves, representing the rate of change in electrical activity, is calculated as the first derivative of the signal around the R-wave peak:

$$Slope_R = \frac{\Delta x}{\Delta t} \tag{12}$$

where Δx is the change in the signal amplitude and Δt is the corresponding time interval.

C. Training the Machine Learning Model:

ECG classification is essential for detecting cardiac abnormalities, with deep learning models using hybrid neural network architecture to automate this process [9]. The PhysioNet database distinguishes four key conditions: hyperkalemia (high potassium, indicated by peaked T-waves), hypocalcemia (low calcium, shown by prolonged QT intervals), normal (no abnormalities), and tachycardia (elevated heart rate, indicated by shortened intervals). ECG signals are preprocessed through normalization, segmentation, and labeling to minimize categorical cross-entropy loss [10]. The model's loss function quantifies the difference between true labels and predicted probabilities for each class to enhance classification accuracy [11]:

$$L = -\sum_{i=1}^{C} y_i log(\hat{y}_i) \tag{13}$$

where y_i is the true label (1 for the correct class, 0 otherwise), and $\hat{y}_i$ is the predicted probability for class i.

IV. HARDWARE IMPLEMENTATION:

The trained ML model was implemented on the PYNQ Z2 FPGA board, a high-performance platform for AI and ML applications. Powered by the Xilinx Zynq-7000 series SoC, the PYNQ Z2 integrates programmable logic (FPGA) and ARM cores, providing a flexible environment for real-time signal processing [12]. The PYNQ framework, which includes Python libraries, enables seamless integration with the FPGA, facilitating efficient execution of the ECG arrhythmia detection model. The board's parallel processing, low latency, and low power requirements make it ideal for portable, real-time ECG monitoring. Its reconfigurable architecture ensures optimized performance and resource efficiency [13, 14].

V. PERFORMANCE METRICS

Performance metrics are quantitative measures to assess an ML model or a system's efficiency and effectiveness. These measurements provide insights into the model's performance in terms of its ability to make accurate predictions, classify instances correctly, and generalize to unseen data [15].

A. AI/ML Model Performance Analysis

Table I shows the metrics used for the AI/ML model performance analysis.

B. FPGA Device Performance Analysis:

Table II describes the metrics used for FPGA device performance analysis.

TABLE I. AI/ML MODEL PERFORMANCE ANALYSIS

	Metric		Description	Equation
1	Confusion matrix	True Positives (TP)	The number of classes correctly predicted as positive.	-
		False Positives (FP)	The number of classes incorrectly predicted as positive when they are actually negative.	-
		True Negatives (TN)	The number of classes correctly predicted as negative.	-
		False Negatives (FN)	The number of classes incorrectly predicted as negative when they are actually positive.	-
2	Accuracy		The percentage of correctly identified examples in the dataset relative to all occurrences.	$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$
3	Precision		The percentage of true positive predictions among all instances predicted as positive.	$Precision = \frac{TP}{TP + FP}$
4	Recall		The percentage of true positive predictions among all real positive examples in the dataset.	$Recall = \frac{TP}{TP + FN}$
5	F1 score		The harmonic mean of precision and recall, providing a balanced evaluation of performance, especially for imbalanced datasets.	$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$

TABLE II. EDGE DEVICE PERFORMANCE ANALYSIS

	Metric	Description	Equation
1	Branch misses	Occurs when the processor's branch prediction mechanism fails to predict the outcome of a branch instruction.	Branch Misses = Total Branch Instructions - Correctly Predicted Branches
2	Bus cycles	The number of cycles the CPU spends waiting for data transfers on the system bus.	Bus Cycles = (Memory Access Time) / (CPU Clock Cycle Time) × Number of Memory Accesses
3	Cache misses	Occurs when data requested by the CPU is not found in the cache memory.	Cache Misses = Total Memory Accesses - Cache Hits
4	CPU cycles	Total cycles executed by the CPU during program runtime.	CPU Cycles = CPU Clock Rate × Program Execution Time
5	Instructions	Total number of instructions executed by the CPU.	Instructions = Sum of all executed instructions
6	Instructions Per Cycle (IPC)	Average number of instructions executed per CPU cycle.	IPC = Instructions / CPU Cycles
7	Time elapsed	Total wall-clock time taken to execute the program.	Time Elapsed = End Time - Start Time
8	Energy	The amount of work a system can do during its operation.	Energy = Power × Time

VI. RESULTS AND DISCUSSION

A. *AI/ML Model Performance Results*

Figure 6 illustrates the performance analysis of the AI/ML model. Figure 6(a) displays the accuracy and classification report, while Figure 6(b) presents the confusion matrix results. The analysis was carried out using the TensorFlow Lite framework and the MobileNetV2 model in Google Colab to evaluate the AI/ML model's performance.

The MIT-BIH Arrhythmia Database was used as the primary dataset for ECG signal processing and analysis. From the dataset, 60% of the data was allocated for training to develop and optimize the model, while the remaining 40% was reserved for testing to evaluate the model's performance.

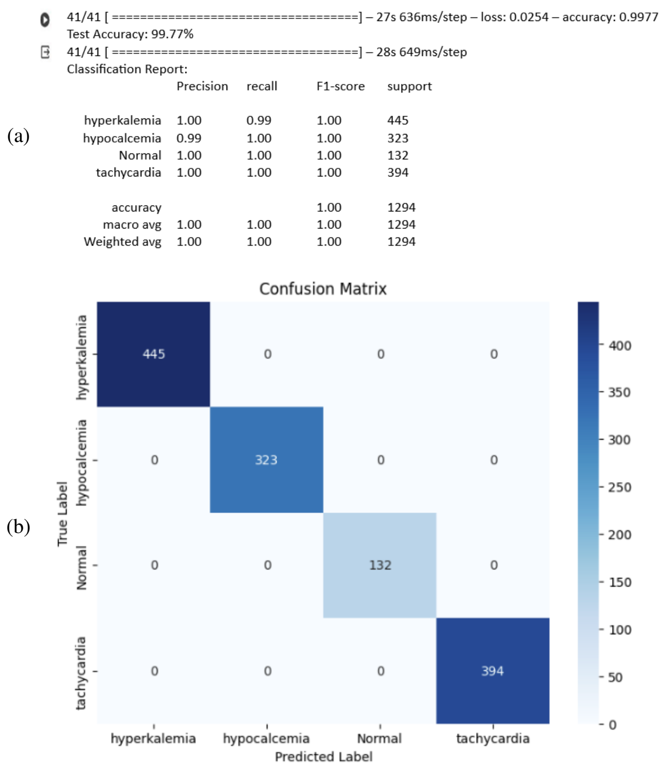

Fig. 6. Performance analysis.

B. *PYNQ FPGA Results*

Table III provides detailed performance profiling metrics by using the PERF profiler for three critical stages: Feature extraction, ECG preprocessing, and classification of ECG signals when executed on the PYNQ FPGA board. The profiling results include key hardware counters such as branch misses, cache misses, CPU cycles, and instructions, which reflect the computational efficiency and memory performance of each process.

TABLE III. EDGE DEVICE PERF PROFILING RESULTS EXECUTED ON PYNQ FPGA

	Feature extraction	ECG preprocessing	ECG classification
Branch misses	1,912,71	1,841,679	1,895,263
Cache misses	1,039,007	1,384,698	1,044,776
Cache references	27,893,441	28,763,364	27,854,907
CPU cycles	141,111,349	160,596,365	141,526,115
Instructions	77,722,026	78,076,671	77,591,158
Time elapsed (s)	0.221188959	0.279711334	0.221537474
IPC	0.55	0.49	0.55

The CPU cycles consumed were approximately 141 million for feature extraction, 160 million for ECG preprocessing, and 141 million for classification. The instructions executed during these processes are approximately 77.7 million, 78 million, and 77.5 million, respectively. The total time elapsed for execution is 0.22 s for feature extraction and classification, while preprocessing takes 0.28 s. The IPC for feature extraction and classification is 0.55, with ECG preprocessing slightly lower at 0.49. PYNQ is more suitable for applications that demand high parallel processing and specialized hardware acceleration, such as real-time signal processing, because of its reconfigurable architecture and ability to optimize for low-power operation in complex computational tasks.

Table IV shows a comparative analysis of the proposed with existing models based on platform, accuracy, processing speed, and power consumption. The methods in [9, 10, 11, 12] and the proposed one achieved accuracy levels of 99.67%, 96%, 99.20%, 86.7%, and 99.7%, while the power consumption was 0.45, 3.3, 2.81, and 0.42 mW, with a processing speed of 15, 0.017, 0.2, 2.875, and 0.723 s, respectively. However, the proposed model strikes the best balance between accuracy, processing speed, and power efficiency, making it the most optimal choice for real-world applications that require high precision and power efficiency.

TABLE IV. COMPARISON OF PROPOSED WITH PREVIOUS APPROACHES

Study	Platform	Accuracy (%)	Processing speed (s)	Power consumption (mW)	Dataset used
[9]	FPGA	99.67	15	0.45	PTB database
[10]	FPGA	96	0.017	3.3	Temporal Convolutional Networks
[11]	FPGA	99.20	0.2	-	PTB diagnostic ECG database
[12]	FPGA	86.7	2.875	2.81	ECG recordings
Proposed model	FPGA	99.7	0.723	0.42	MIT-BIH Arrhythmia Database

VII. CONCLUSION

This study presents an efficient, real-time FPGA-based system for ECG arrhythmia detection, integrating ML and fiducial windowing techniques. The system utilizes Keras and TensorFlow Lite for the ML model, allowing robust classification with high accuracy. The proposed system was implemented on the PYNQ FPGA board, leveraging its parallel processing capabilities for low latency and power efficiency. Fiducial windowing enhances feature extraction accuracy by focusing on critical ECG components, while a Butterworth filter reduces noise, and DWT extracts essential features efficiently. The PERF profiler was used for performance analysis, evaluating system metrics such as execution time.

The proposed model achieved a classification accuracy of 99.7%, with a processing speed of 0.723 s and a power consumption of 0.42 mW. This balance outperforms the existing methods in [9, 10, 11, 12], which, despite achieving accuracies of 86.7-99.67%, suffer from higher power consumption or slower speeds. Although GPU-based solutions can deliver faster processing, their power consumption can reduce energy efficiency. The proposed model excels with its superior accuracy, optimized energy efficiency, and moderate speed, making it the most practical solution for real-world low-power applications. Future studies could explore CNNs, RNNs, and hybrid ML approaches for enhanced ECG analysis in resource-constrained environments.

ACKNOWLEDGMENT

The authors warmly acknowledge the VGST, Govt. of Karnataka, and REVA University, Bengaluru, Karnataka, for providing the facilities required to carry out the research. They also acknowledge the support extended by VGST, Govt. of Karnataka, in sponsoring this research with software tools and project no. KSTePS/VGST/GRD-684/KFIST(LI)/2018. They also acknowledge the School of ECE REVA University for supporting this work.

REFERENCES

[1] G. Aishwarya *et al.*, "A Survey on use of FPGA in Automotive System," in *2022 International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, Jul. 2022, pp. 51–56, https://doi.org/10.1109/DISCOVER55800.2022.9974941.

[2] D. Suresh, P. V. Joshi, P. Parandkar, and K. M. Sudharshan, "An Improved Authentication Scheme for V2I Communication," *SN Computer Science*, vol. 5, no. 5, May 2024, Art. no. 535, https://doi.org/10.1007/s42979-024-02865-7.

[3] P. V. Joshi and K. S. Gurumurthy, "Analysing and improving the performance of software code for Real Time Embedded systems," in *2014 2nd International Conference on Devices, Circuits and Systems (ICDCS)*, Combiatore, India, Mar. 2014, pp. 1–5, https://doi.org/10.1109/ICDCSyst.2014.6926134.

[4] A. Biran, "Ecg Bio-Identification Using Fiducial and Non-Fiducial Techniques: A Proposed Methodology Based on Short Time Fourier Transform and Histograms of Qrs Features." Social Science Research Network, May 24, 2022, https://doi.org/10.2139/ssrn.4118100.

[5] G. B. Moody, R. G. Mark, and A. L. Goldberger, "PhysioNet: a research resource for studies of complex physiologic and biomedical signals," in *Computers in Cardiology 2000. Vol.27 (Cat. 00CH37163)*, Cambridge, MA, USA, 2000, pp. 179–182, https://doi.org/10.1109/CIC.2000.898485.

[6] F. Bagheri, N. Ghafarnia, and F. Bahrami, "Electrocardiogram (ECG) Signal Modeling and Noise Reduction Using Hopfield Neural Networks," *Engineering, Technology & Applied Science Research*, vol. 3, no. 1, pp. 345–348, Feb. 2013, https://doi.org/10.48084/etasr.243.

[7] K. P. Nandini and G. Seshikala, "Smart IoT Solutions for Precise COVID-19 Lung and Heart Disease Identification via Bayesian Neural Networks and Hybrid Optimization," *SN Computer Science*, vol. 5, no. 5, May 2024, Art. no. 510, https://doi.org/10.1007/s42979-024-02842-0.

[8] K. P. Nandini and G. Seshikala, "Role of Embedded Computing Systems in Biomedical Applications–Opportunities and Challenges," in *2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, Nitte, India, Nov. 2021, pp. 332–335, https://doi.org/10.1109/DISCOVER52564.2021.9663646.

[9] V. B. K. L. Aruna, E. Chitra, and M. Padmaja, "Accelerating deep convolutional neural network on FPGA for ECG signal classification," *Microprocessors and Microsystems*, vol. 103, Nov. 2023, Art. no. 104939, https://doi.org/10.1016/j.micpro.2023.104939.

[10] M. Carreras, G. Deriu, L. Raffo, L. Benini, and P. Meloni, "Optimizing Temporal Convolutional Network Inference on FPGA-Based Accelerators," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 3, pp. 348–361, Sep. 2020, https://doi.org/10.1109/JETCAS.2020.3014503.

[11] A. Giorgio, C. Guaragnella, and M. Rizzi, "FPGA-Based Decision Support System for ECG Analysis," *Journal of Low Power Electronics and Applications*, vol. 13, no. 1, Mar. 2023, Art. no. 6, https://doi.org/10.3390/jlpea13010006.

[12] S. Ran *et al.*, "Homecare-Oriented ECG Diagnosis With Large-Scale Deep Neural Network for Continuous Monitoring on Embedded Devices," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022, https://doi.org/10.1109/TIM.2022.3147328.

[13] R. Srivastava, B. Kumar, F. Alenezi, A. Alhudhaif, S. A. Althubiti, and K. Polat, "Automatic Arrhythmia Detection Based on the Probabilistic Neural Network with FPGA Implementation," *Mathematical Problems in Engineering*, vol. 2022, no. 1, 2022, Art. no. 7564036, https://doi.org/10.1155/2022/7564036.

[14] Y. Vasa, S. Reddy, and J. V. Suman, "Automated Machine Learning Framework Using Large Language Models for Financial Security in Cloud Observability," *International Journal of Research and Analytical Reviews*, vol. 9, no. 3, pp. 181–189, 2022.

[15] S. M. Shaaban, M. Nawaz, Y. Said, and M. Barr, "An Efficient Breast Cancer Segmentation System based on Deep Learning Techniques," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12415–12422, Dec. 2023, https://doi.org/10.48084/etasr.6518.