Performance and Interpretability Trade-offs in Mixed-Precision and Quantized ECG Arrhythmia Classifiers for Edge Deployment

Nikhilesh Kancherla, Chavan Rupesh Sharan, Cooly Siddarttha, Ajay Bhaskar Reddy Satti, Dr.Kirithiga Nandini

Computer Science and Engineering, SRM Institute of Science and Technology Chennai , India

Abstract— **Continuous ECG monitoring is becoming more and more important for finding arrhythmias early, especially in wearable and edge-based healthcare systems where memory, power, and latency are all limited. This project looks at how different deep learning models act when they are optimized for deployment using mixed precision and quantization, while still making sure that their predictions are accurate. This work uses the MIT-BIH Arrhythmia Database to compare the 1D-CNN, ResNet1D, BiLSTM, and Transformer1D models in FP32, FP16/BF16, and INT8 settings using Post-Training Quantization and Quantization-Aware Training. The project looks at both classification performance and interpretability. It uses Saliency, SHAP, and LIME to see if compressed models still focus on important ECG areas like the QRS complex and strange beat patterns. The study is set up as a full pipeline with preprocessing, training, optimization, ONNX export, and explanation analysis. This makes it good for real-time edge deployment. In general, the work shows that model compression can greatly improve efficiency. However, the best practical balance is reached when accuracy, latency, and explanation stability are all taken into account at the same time.**

Keywords—* *ECG arrhythmia classification, edge deployment, mixed precision, quantization, explainable AI, MIT-BIH arrhythmia database, interpretability drift.

I. INTRODUCTION

Despite the variety of causes of deaths worldwide, cardiovascular diseases remain to be one of the most widespread ones, and therefore early identification of heart issues plays a significant role in healthcare. Electrocardiogram (ECG) signals provide valuable data concerning the functioning of the heart electrical activity and can be used in searching arrhythmias. With the increased usage of wearable technology and remote monitoring devices, automated and real-time ECG analysis directly on edge devices increasingly becomes a necessity.

More recent advancements in artificial intelligence and specifically deep learning have enabled ECG classification to be more accurate. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based systems can extract complex patterns on ECG signals, and thus the various forms of arrhythmias can be reliably detected. These techniques have been effective in research context and may enable physicians to make judgements at the clinic.

Nevertheless, it is not easy to implement this type of models into real healthcare systems. A majority of the deep learning models that are effective require much memory and processing and are highly computationally intensive. This renders their application to platforms with scarce resources, such as wearable devices and edge-based monitoring systems with power consumption, latency, and storage, all highly considered.

In order to address these issues, mixed precision computation and quantization, the methods of model optimization, are gaining popularity. They reduce the numerical precision and simplify calculations and consume less memory, thus accelerating the inference and improving the performance of deployment. Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT) are two methods that can be used to make deep learning models work in real time. However it may be accompanied with some tradeoffs in the form of precision and stability of models.

Interpretability, as well as performance, is equally crucial in medicine. An effective healthcare paradigm has to provide accurate forecasts and explain understandable reasons to its decisions. This plays a crucial role in the development of trust between the healthcare professionals and ensuring that the technology can be deployed in practice. Saliency Maps, SHAP and LIME are representative explainable AI techniques that can assist you to visualize what aspects of the ECG signal contribute to model predictions.

Within this, the present research will focus on the development of a viable and understandable system of ECG arrhythmia classification that can be deployed to the edge. The paper evaluates several deep learning networks that are developed with different optimization strategies, such as mixed-precision and INT8 quantization. It also examines the impact of these optimizations not only on performance measures such as performance in terms of accuracy and latency, but also on the consistency in model explanations.

The objective of this work is to offer a practical and balanced solution in real-time ECG monitoring systems in contemporary medical settings by incorporating model efficiency, predictive performance and interpretability analysis into a single framework.

II. LITERATURE SURVEY

Recent studies have looked into using deep learning and edge-oriented methods to classify ECG arrhythmias. Xiaolin et al. [1] put forward a two-stage ECG classification system for decentralized inferencing along the edge–cloud continuum. Their method worked well and had low latency, but it used distributed processing instead of a fully compact edge model.

Farag et al. [2] also came up with an STFT-based CNN model for classifying ECGs at the edge. Their approach enhanced feature representation and facilitated quantization for deployment; however, it did not evaluate interpretability or model reasoning.

Liu et al. [3] created an ECG classification processor that uses very little energy for wearable monitoring. Their design cut down on power use a lot while still being very accurate, but it was more focused on optimizing hardware than on making model comparisons more flexible.

Bechinia et al. [4] suggested a light CNN with a fine-tuned MobileNetV2 for classifying ECGs. Their method worked well on datasets that weren't balanced and could be used in real time, but it didn't have good temporal modeling capabilities.

Chen et al. [5] introduced a hybrid weak–strong classifier that integrates linear methods and SVM for the detection of arrhythmia. Their method cut down on the cost of computing, but it didn't use deep learning to get advanced features.

Dekimpe et al. [6] executed ECG classification on an ultra-low-power microcontroller. Their system showed that it could work in real time, but it didn't look into ways to compress models, like quantization-aware training.

Chandra et al. [8] put forward a smart biomedical sensor network that uses autoencoders and decision trees to keep an eye on more than one patient at once. Their system could detect things from different places, but it didn't have deep neural networks for learning complex patterns.

Bayasi et al. [9] created a low-power ECG processor that used fiducial features and naive Bayes classification. Their method worked well, but it relied a lot on manual feature extraction.

Hou et al. [10] suggested using an LSTM-based autoencoder with SVM to classify ECGs. Their method effectively captured temporal dependencies, but it made the calculations more complicated.

Janveja et al. [11] presented a low-power co-processor for predicting ventricular arrhythmia. Their design was able to make early predictions while using less power, but it mostly focused on making the hardware work better.

Cai et al. [12] introduced an event-driven artificial neural network core for the classification of arrhythmias in wearable sensors. Their system used less power, but it didn't help with model generalization across datasets.

Pal et al. [13] suggested a two-stage classifier that uses both morphological features and deep learning. Their method made things more efficient, but it didn't look at different ways to compress data.

The studies above show that most current systems are either trying to make things more accurate, use less power, or work better at the edge. But not many methods look at both model compression and how easy it is to understand at the same time.

To get around these problems, the proposed system uses a unified approach that combines different deep learning architectures with mixed precision, quantization techniques, and explainable AI methods. This makes it possible to deploy at the edge quickly while still making accurate and easy-to-understand predictions.

III. PROPOSED WORK

The proposed system provides a scalable and efficient structure on the classification of ECG arrhythmias through a combination of deep learning, model compression algorithms, and explainable AI. The goal of this system is to identify a compromise between performance, efficiency and interpretability to edge deployment, unlike traditional approaches that emphasize only on smoothing out accuracy or reducing the computational cost.

The dataset used in the system is the MIT-BIH Arrhythmia and a structured preprocessing pipeline that incorporates filtering noise and reduction, signal splitting, and normalization. Such will ensure that the ECG signals are clean, standardized and suitable to train the model making the overall classification performance even better.The system uses the MIT-BIH Arrhythmia dataset and a structured preprocessing pipeline that includes filtering out noise, breaking up signals, and normalizing them. These steps make sure that the ECG signals are clean, standardized, and good for training the model, which makes the overall classification performance better.

Unlike other works that use just one model, the proposed system uses several deep learning architectures, such as 1D-CNN, ResNet1D, BiLSTM, and Transformer-based models. This multi-model approach lets you compare different architectures based on how accurate they are, how fast they are at computing, and how well they work in real-time applications.

The system uses model optimization techniques like mixed precision (FP16/BF16) and quantization methods like Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT) to solve deployment problems. These methods cut down on memory usage and inference time by a lot, which makes the models good for edge devices that don't have a lot of resources.

The proposed work's main feature is the use of explainable AI methods like SHAP, LIME, and Saliency Maps. These methods help us understand model predictions better by showing us important parts of the ECG signal. This makes medical decision-making more open and trustworthy.

This system also introduces the idea of interpretability drift, which looks at how model explanations change after using compression methods. This is different from earlier studies. This makes sure that optimized models not only stay accurate but also keep their ability to reason in a meaningful way.

The system also supports deployment through ONNX conversion and has a Streamlit-based interface for making predictions and visualizing them in real time. This makes it easy to use, compare performance, and interact with healthcare apps in real life.

The proposed system thus offers a holistic solution that integrates precision, efficacy, and clarity, rendering it exceptionally appropriate for practical ECG monitoring and edge-based healthcare systems.

A. System Architecture

The proposed system's overall architecture is a modular pipeline that combines data processing, model training, optimization, and deployment to make it easier to classify ECG arrhythmias. The system makes sure that the flow from getting raw ECG signals to making predictions and interpretations is smooth. This makes it good for real-time edge-based applications.

The data acquisition and preprocessing module is the first part of the architecture. It records and analyzes ECG records of MIT-BIH Arrhythmia dataset. This stage includes noise reduction, division of heartbeats into R-peaks and equalization of signal values. These pre-processing steps enhance a signal quality and also ensure that all the models are fed on the same input.

The processed information is then forwarded to the model training module that contains a number of deep learning models, such as 1D-CNN, ResNet1D, BiLSTM, and Transformer models. The models are separately trained to learn various aspects of ECG signals, such as their spatial properties, their temporal independent variables, and scale relationships.

The architecture also has a model optimization unit, post-training, enableing the mixed-precision (FP16/BF16) and quantization strategies, such as Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT). The step causes the model to become smaller and simplified to compute and makes it viable to execute it at devices with limited resources.

After training, the architecture has a model optimization module that uses mixed precision techniques (FP16/BF16) and quantization methods like Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT). This step makes the model smaller and less complicated to compute, making it possible to run it efficiently on devices with limited resources.

After being optimized, the models are sent to the deployment module, where they are changed to ONNX format so that they can work on more than one platform and make predictions faster.This lets the system act like it is deploying edges in the real world and measure things like latency and memory usage.

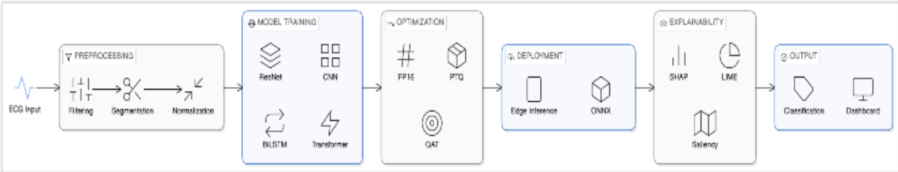

Fig.1 System Architecture

Fig. 1 shows the system architecture of the proposed ECG arrhythmia classification framework. The system begins with ECG input signals that are processed by a preprocessing module that includes filtering, segmentation, and normalization.

The processed data is then sent to the model training module, which has CNN, ResNet, BiLSTM, and Transformer models that can extract features and sort them. To make the trained models work better, they are further optimized using methods like FP16, PTQ, and QAT.

We use ONNX to deploy the optimized models for edge inference. To help people understand what the model is predicting, there is an explainability module that uses SHAP, LIME, and saliency maps. The output module finally gives arrhythmia classification results and a dashboard for viewing them.

The proposed architecture will thus provide a holistic end-to-end pipeline that incorporates preprocessing, multi-model learning, optimization, deployment, and explainability making it highly effective in real-life ECG monitoring and edge-based healthcare applications.

IV. METHODOLOGY

The proposed system uses a structured method that combines data preprocessing, training a deep learning model, optimization, and interpretability analysis to accurately and quickly classify ECG arrhythmias. The methodology is made to make sure that all models are the same and that they can be fairly compared and ready for use.

The first step is to get the data and clean it up. This is where the ECG signals come from the MIT-BIH Arrhythmia dataset. Noise filtering is done on the raw signals to get rid of baseline wander and high-frequency interference. Next, R-peak detection is used to break the ECG signals down into separate heartbeats. Then, normalization is used to make sure that amplitude changes are all the same. These steps make sure that the data that goes into the model is clean and consistent.

In the next step, the preprocessed ECG signals are used to train several deep learning models. The models are made up of 1D Convolutional Neural Networks (1), ResNet1D, BiLSTM, and Transformer-based architectures. Each model is meant to pick up on different aspects of ECG signals, like their shape, how they change over time, and how they relate to other signals over long distances. First, the models are trained with full precision (FP32) to set a baseline for performance.

$$y(t) = \sum_{i=0}^{k-1} x(t+i) * w(i) + b \tag{1}$$

Where:
x = input signal
w = filter/kernel
b = bias
k = kernel size

After training the model, optimization methods are used to make it more efficient for deployment on the edge. To save memory and speed up training, mixed precision training with FP16/BF16 is used. Also, quantization methods like Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT) are used to change models into lower precision formats (INT8), which makes the calculations easier and faster.

The methodology includes a step for analyzing interpretability to see how clear the model is. Saliency Maps, SHAP, and LIME are examples of explainable AI methods that can be used to find important parts of the ECG signal that affect model predictions. This step makes sure that the models are not only correct but also give useful information for making medical decisions.

Lastly, some performance metrics are utilized to test the system, which include: memory usage, latency, and accuracy. In order to strike the optimal solution between performance and efficiency, comparisons of various models and optimization techniques are made. The process of using ONNX to optimize the models is followed by the application of the models that are optimized to perform a real-time inference and display the results using a Streamlit interface.

The proposed approach, in turn, represents an entire framework encompassing signal processing, deep learning, model optimization and explainability to obtain an effective and reliable ECG classification system that can be used in the real world.

V. IMPLEMENTATION

The proposed system is developed using python and modular architecture that allows you to pre process data, model, train, optimize and evaluate them and deploy them. Its implementation is supposed to be flexible, scalable, as well as simple to test on various deep learning models and optimization techniques.

The preprocessing stage is accomplished in dedicated scripts which filter, segment and normalize ECG signals. We use R-peak detection to find individual heartbeat segments in the raw MIT-BIH dataset. We also standardize the signals such that they match in all the patient records. The processed data is then stored in an orderly manner in order to train and test fast.

It is developed with the PyTorch framework and various architectures such as 1D-CNN, ResNet1D, BiLSTM, and Transformer models are assembled as a separate module. Each model has settings which you can change and so you can explore different hyperparameters and settings.

A core training module has the responsibility to load datasets, process batches, compute losses and optimize. We use full precision (FP32) to train baseline models, and we deploy automatic mixed precision to train with mixed precision more efficiently. The training pipeline ensures that model testing is done similarly.

Separate modules that do Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT) are used to optimize the model. These modules transform trained models into lower-precision formats such as INT8, resulting in smaller models and faster inference. The maximized models are then shipped in the ONNX format, in order to be easily run on alternative platforms.

Libraries that implement the interpretability module include SHAP and LIME, and gradient-based methods, including saliency maps. These are the ways we see how model predictions match or diverge, and can identify key components in the ECG signal, which gives us more insight into model behavior.

The evaluation module figures out important performance metrics like accuracy, precision, recall, F1-score, latency, and memory usage. A comparative analysis is done to see how different models and optimization methods affect both performance and efficiency.

Lastly, the system has a user interface made with Streamlit that lets users enter ECG signals, see predictions, compare model outputs, and see how well the model works in real time. This interface makes the system easier to use and shows how useful it is in real life.

The implementation provides a full end-to-end pipeline that includes preprocessing, training multiple models, optimization, making the system easy to understand, and deployment. This makes the system good for real-world ECG monitoring and healthcare applications that run on the edge.

VI. RESULTS AND ANALYSIS

The efficiency of the suggested system is explored with different accuracy dropouts using various deep learning models such as FP32, FP16 and INT8-based quantization plans. The evaluation scales of measurement will be Accuracy, Precision, Recall and F1 Score. Table I shows the results of the comparison.

TABLE I. Overall Models Performance Metrics

Model	Precision Format	Accuracy	Precision	Recall	F1 Score
BiLSTM	FP16	0.9580	0.7560	0.9582	0.8245
BiLSTM	FP32	0.9491	0.7340	0.9543	0.7979
CNN1D	FP16	0.9645	0.7781	0.9690	0.8367
CNN1D	FP32	0.9420	0.7278	0.9606	0.7830
CNN1D	INT8 Dynamic	0.9415	0.7268	0.9607	0.7822

CNN1D	INT8 Static	0.9453	0.7354	0.9608	0.7893
CNN1D	QAT INT8	0.9752	0.8432	0.9648	0.8931
ResNet1D	FP16	0.9727	0.8110	0.9728	0.8668
ResNet1D	FP32	0.9639	0.7844	0.9646	0.8461
ResNet1D	INT8 Dynamic	0.9637	0.7834	0.9646	0.8452
Transformer	FP16	0.9384	0.7172	0.9581	0.7823
Transformer	FP32	0.9458	0.7235	0.9547	0.7893

As Table I indicates, the classification effectively depends on the various model architecture and precision formats. Most architectures have mixed precision (FP16) that always improves model performance over full precision (FP32). The BiLSTM model, e.g., achieves an F1 score of 0.9491 (FP32) and then 0.9580 (FP16) indicating that the model is more precise.

The CNN1D architecture QAT INT8 is the superior amongst all the models having an accuracy of 0.9752 and F1 score of 0.8931. This implies that QAT preserves the accuracy of the model even following the quantization, and it makes it a superb option in cases of edge deployment.

You can compare PTQ (INT8 Dynamic and INT8 Static) methods to the performance of QAT to observe that it causes a slight decrease in the performance. An example of CNN1D INT8 Dynamic is 0.9415 accuracy which is less than the CNN1D FP16. But this is much improved by QAT which modifies the model in training so as to cope with quantization effects.

All precision formats also can be used with ResNet1D. For example, FP16 gets an accuracy of 0.9727 and an F1 score of 0.8668. Further, the difference between the FP32 and the INT8 Dynamic in ResNet1D is very small, which proves that the design is capable of sustaining the quantization.

The Transformer model, on the other hand, doesn't work as well as CNN1D and ResNet1D. FP32 makes FP16 a bit less precise when it comes to the Transformer, yet the F1 score is lower over all. This implies that Transformer-based models could use further enhancement to be more accurate in the classification of ECG signals..

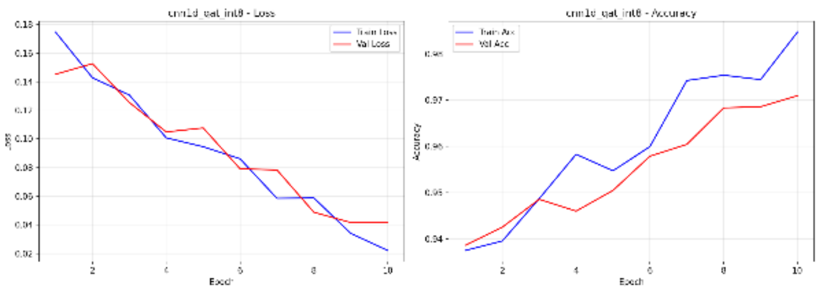

Fig. 2 Training and Validation accuracy and loss curves

Fig. 2 demonstrates the success of CNN1D with QAT INT8 in training and testing with the goal of minimizing loss by the end of training, as well as achieving good test performance. The loss decreases continuously and this is an indication that learning and convergence are taking place. The accuracy continues to increase, the training accuracy is approximately 0.985 and the validation accuracy is approximately 0.97. The slight margin between them indicates that the model is robust and can be generalized.

Overall, the findings indicate that optimization procedures such as the mixed precision, and quantization employed in models can significantly enhance efficiency without reducing the accuracy. The most appropriate method to achieve a balance between performance and computational efficiency is by training quantization-Aware. This is what makes it ideal in real time applications of healthcare running on the edge.

VII. CONCLUSION AND FUTURE WORK

The proposed system is an excellent means to categorize ECG arrhythmias as it involves mixed precision and quantization, as well as multiple deep learning models. The results show that FP16 works better than FP32, and Quantization-Aware Training (QAT) is the best way to find the right balance between accuracy and efficiency. The CNN1D QAT INT8 model is the most effective, hence can be deployed to the edge. Incorporating explainable AI techniques also ensures that predictions can be correct and intelligible.

This system can be scaled to consider several datasets in order to increase generalization in future activities. It is possible to test real time practical performance of wearable devices. Lightweight hybrid models and better ways to explain things can also be made to make accuracy and clinical usability even better.

REFERENCES

[1] L. Xiaolin, B. Cardiff, and D. John, "A Two-Stage ECG Classifier for Decentralized Inferencing Across Edge-Cloud Continuum," IEEE Sensors Journal, vol. 24, no. 14, pp. 23324–23336, 2024.

[2] M. M. Farag, "A Self-Contained STFT CNN for ECG Classification and Arrhythmia Detection at the Edge," IEEE Access, vol. 10, pp. 12345–12356, 2022.

[3] J. Liu et al., "A High-Accuracy and Ultra-Energy-Efficient Cardiac Arrhythmia Classification Processor for Wearable Intelligent ECG Monitoring," IEEE Journal of Solid-State Circuits, vol. 60, no. 10, pp. 3762–3775, 2025.

[4] H. Bechinia, D. Benmerzoug, and N. Khlifa, "Lightweight Custom CNN and Fine-Tuned MobileNetV2 for ECG Arrhythmia Classification," IEEE Access, vol. 12, pp. 45678–45689, 2024.

[5] Z. Chen et al., "An Energy-Efficient ECG Processor With Weak–Strong Hybrid Classifier for Arrhythmia Detection," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 65, no. 7, pp. 948–952, 2018.

[6] R. Dekimpe and D. Bol, "ECG Arrhythmia Classification on an Ultra-Low-Power Microcontroller," IEEE Transactions on Biomedical Circuits and Systems, vol. 16, no. 3, pp. 456–467, 2022.

[7] N. Kumar and S. Raj, "An Adaptive Scheme for Real-Time Detection of Patient-Specific Arrhythmias Using Single-Channel Wearable ECG Sensor," IEEE Sensors Journal, vol. 8, no. 3, pp. 7001504, 2024.

[8] S. Chandra and R. Gupta, "Smart Biomedical Sensor Network for Multipatient Cardiac Arrhythmia Monitoring," IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 4000209, 2023.

[9] N. Bayasi et al., "Low-Power ECG-Based Processor for Predicting Ventricular Arrhythmia," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 5, pp. 1962–1972, 2016.

[10] B. Hou, J. Yang, P. Wang, and R. Yan, "LSTM-Based Auto-Encoder Model for ECG Arrhythmias Classification," IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 4, pp. 1232–1242, 2020.

[11] M. Janveja et al., "A Low-Power Co-Processor to Predict Ventricular Arrhythmia for Wearable Healthcare Devices," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 32, no. 9, pp. 1672–1683, 2024.

[12] Q. Cai et al., "A 1.3 μW Event-Driven ANN Core for Cardiac Arrhythmia Classification in Wearable Sensors," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 9, pp. 3123–3127, 2021.

[13] D. Pal, S. Mukhopadhyay, and R. Gupta, "Two-Stage Classifier for Resource-Constrained On-Board Cardiac Arrhythmia Detection," IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 2502810, 2023.

[14] A. Nasim et al., "An Evolutionary-Neural Mechanism for Arrhythmia Classification With Optimum Features Using Single-Lead ECG," IEEE Access, vol. 10, pp. 56789–56800, 2022.

[15] J. Huang et al., "ECG Arrhythmia Classification Using STFT-Based Spectrogram and Convolutional Neural Network," IEEE Access, vol. 7, pp. 98765–98775, 2019.