ECG-based real-time arrhythmia monitoring using quantized deep neural networks: a feasibility study

ECG-based Real-time Arrhythmia Monitoring Using Quantized Deep Neural Networks: A Feasibility Study

Henrique De Melo Ribeiro[1,*], Ahran Arnold[2], James P Howard[2], Matthew J Shun-Shin[2], Ying Zhang[1], Darrel P Francis[2], Phang B Lim[2], Zachary Whinnett[2], Massoud Zolgharni[1,2]

[1]School of Computing and Engineering, University of West London, UK
[2]National Heart and Lung Institute, Imperial College London, UK

*Corresponding author: email address: henrique.demeloribeiro@uwl.ac.uk

ABSTRACT

Continuous ambulatory cardiac monitoring plays a critical role in early detection of abnormality in at-risk patients, thereby increasing the chance of early intervention. In this study, we present an automated ECG classification approach for distinguishing between healthy heartbeats and pathological rhythms. The proposed lightweight solution uses quantized one-dimensional deep convolutional neural networks and is ideal for real-time continuous monitoring of cardiac rhythm, capable of providing one output prediction per second. Raw ECG data is used as the input to the classifier, eliminating the need for complex data preprocessing on low-powered wearable devices. In contrast to many compute-intensive approaches, the data analysis can be carried out locally on edge devices, providing privacy and portability. The proposed lightweight solution is accurate (sensitivity of 98.5% and specificity of 99.8%), and implemented on a smartphone, it is energy-efficient and fast, requiring 5.85mJ and 7.65ms per prediction, respectively.

Keywords: arrhythmia, ECG, deep learning, continuous monitoring, heart disease

1. INTRODUCTION

According to the World Health Organization, cardiovascular diseases are the cause of death for 17.9 million people yearly, totaling 31% of global deaths, where 80% of premature heart diseases are preventable [1]. ECG provides rich information regarding the patient's heart activity, enabling the diagnosis of a variety of cardiac abnormalities ranging from acute coronary syndrome to arrhythmias [2]. ECG monitoring in daily life is, therefore, a necessary way for early diagnosis of heart disease.

The conventional methods of long-time ECG monitoring are, however, invasive and expensive, and it hinders the daily activity of the patients. Additionally, heart activity is usually recorded for 24 to 48 hours. Therefore, unless the heart condition manifests within this period, the results may be inclusive.

Hardware and software advancements have led to the development of wearable devices (e.g wristwatches, patches, smartphones, and vests), which are both practical and affordable, and enable continuous monitoring of cardiac rhythms in vulnerable populations from the comfort of their homes, while at the same time providing critical alerts for events requiring prompt medical attention or hospitalization [3].

The collected ECG data can be analyzed in two ways (figure 1); cloud-based and local. In cloud-based solutions, the ECG signal is usually transmitted using wireless transmission techniques such as Wi-Fi or Bluetooth. The data is then analyzed on the cloud and feedback is sent back to the user or care providers.

In this study, however, our approach was to locally process the ECG signal by executing the arrhythmia classifier on patients' personal smartphones. The benefit of the processing is three-fold: it allows for continuous operation regardless of the network availability/speed; it can provide an immediate alarm to the patient in case anomalies are suspected; it allows avoiding privacy issues of cloud processing by keeping the patients' data on their personal devices.

However, continuous monitoring on wearable devices requires the ECG classification algorithm to be optimized for the competing objectives of accuracy and low energy consumption. In this study, therefore, our aim was to propose an energy-efficient and accurate deep neural network model to detect arrhythmia anomalies for energy-constrained platforms such as wearable devices.

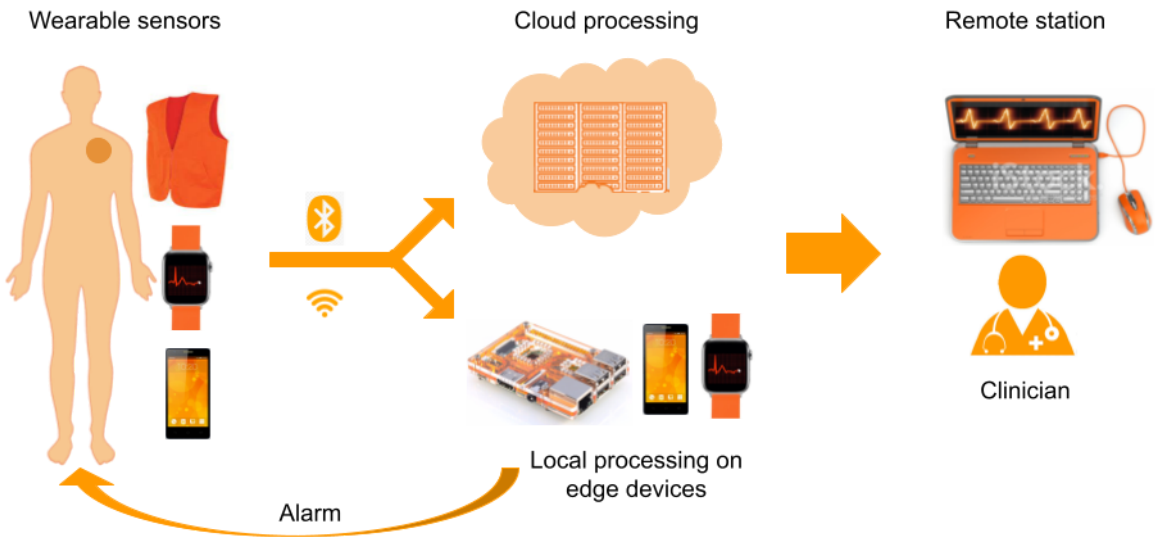


Figure 1. Concept of continuous ambulatory cardiac monitoring using wearable technology.

1.1. Related Work

Traditional machine learning methods use various hand-engineered features to obtain representations of input data. Since the ECG waveform and its morphological characteristics, such as the shapes of QRS complex and P waves, significantly vary under different circumstances and for different patients, the fixed features employed in such algorithms are not sufficient for accurately distinguishing among different types of arrhythmia for all patients [4].

As the feature extraction process is automated with convolutional neural networks (CNNs), the use of CNN has become widespread in this field. These networks are used to classify patient-specific beats [5] and long duration ECG signals containing multiple rhythm classes [6], to detect different interval ECG segments [7], different types of ECG beats [8], and atrial fibrillation [9].

Most related works have aimed at enhancing the accuracy of heartbeats classification [5,8,10], and focused less on the issues of energy consumption. Here, we have mainly focused on the studies aimed at designing low-cost solutions for continuous monitoring using wearable sensors.

An ECG monitor system based was reported where the ECG signal is transmitted to an Android phone and then be forwarded to a remote server [11]. Using a PC, the clinicians can view the ECG after logging in to the server. The addition of an alarm program was also proposed via SMS or email [12]. Another ECG monitoring and alarming system were proposed [13], where an Android phone detects alarms that come from the ECG device and sends them to the cloud alarm

server. However, the system lacked any predictive capabilities, assisting the clinicians in diagnosing the heart condition.

An IoT-based low-cost ECG monitoring system was developed in which Raspberry Pi 2 was used for signal processing. Real-time data plotting for visual inspection by the physicians was achieved using a cloud system [14].

An automatic wearable ECG classification and monitoring system was reported which would require preprocessing the ECG data using a stack denoising autoencoder, and then beat detection [15]. A wireless sensor device was used to retrieve ECG data and send it to a Bluetooth 4.2 computer. The same authors later adopted CNNs and active learning to improve their classification performance [16].

Another energy-efficient ECG monitoring for wearable devices was also proposed which again would require several steps of data preprocessing such as denoising, R-peak detection, and heartbeat segmentation [17]. Long short-term memory recurrent neural networks [18], and spiking neural networks [19,20] have also been proposed for energy-efficient ECG classification.

1.2. Main Contributions

In this study, we propose a quantized convolutional neural network designed to reliably classify arrhythmia using the ECG signals, while being lightweight enough for implementation on low-powered edge devices and achieving short prediction (inference) time. Unlike most existing

works that require complex preprocessing methods, such as denoising [21], isolation of the heartbeats using R–R intervals [22], and wavelets [23], raw ECG signals are used directly as input to the arrhythmia classifier, removing the requirement of all preprocessing steps.

We examined the feasibility of the real-time continuous monitoring of the heart by conducting a simulated scenario. Where feasible, we have provided a performance comparison between our proposed approach and the existing work highlighted in the previous section.

2. DATASET

2.1. MIT-BIH Arrhythmia Database

In this study, the publicly available PhysioNet MIT-BIH ECG Arrhythmia Database was used [24]. The database contains 48 half-hour excerpts of ambulatory ECG recordings, obtained from 47 subjects. Twenty-three recordings were chosen at random from a set of 4000 24-hour ambulatory ECG recordings collected from a mixed population of inpatients The remaining 25 recordings were selected from the same set to include less common but clinically significant arrhythmias that would not be well-represented in a small random sample.

The recordings were digitized at 360 samples per second. Two or more cardiologists independently annotated each record, resulting in reference annotations for each beat; with approximately 110,000 annotations in the entire database. The annotations are divided into 5 categories as per guidelines of the Association for the Advancement of Medical Instrumentation (AAMI) EC57 standard in 1998 [25]. Table 1 shows examples of each category.

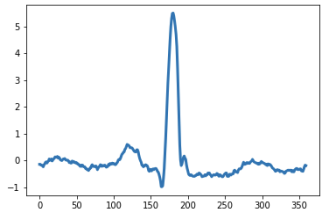
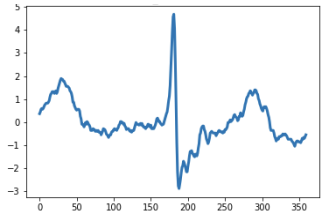
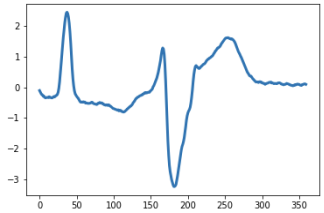
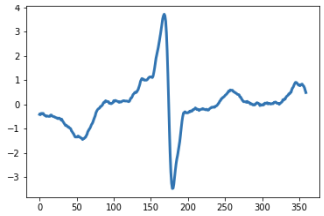
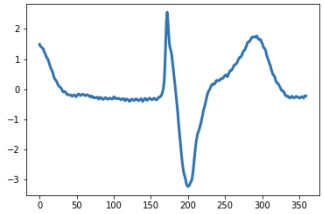
Category	AAMI Classes	Annotation	ECG Sample
N	Normal Beat	Normal Left/Right bundle branch block Atrial escape Nodal escape	
S	Supra-Ventricular Beat	Atrial premature Aberrant atrial premature Nodal premature Supra-ventricular premature	
V	Ventricular Ectopic Beat	Premature ventricular contraction Ventricular escape	
F	Fusion of Ventricular and Normal	Fusion of ventricular and normal	
Q	Unclassifiable Beat	Paced Fusion of paced and normal Unclassifiable	

Table 1. AAMI recommended classes and corresponding ECG signal samples.

2.2. Data Organization

In each recording, the first channel is the modified-lead II (MLII), and the second is fixed as one of V1, V2, V4, and V5 depending on the recording. Since MLII is available in all recordings and considering the fact it has been shown that the use of this lead would be sufficient to achieve high accuracy [26], we, therefore, adopted MLII data in this study.

In order to make the processing feasible with limited computational resources (e.g., edge computing), the long recordings were fragmented into time windows of 1s, thus each containing a fixed length of 360 data points. Based on the cardiologist annotations assigned to the original ECG, each time window was labeled as follows: if the entire segment was annotated as one class, the time window was assigned to that class; if there was a change in the expert annotation within the 1s time-window, then the predominant class was assumed as the label for the entire segment. Examples of the ECG segments are shown in Table 1. This resulted in a total number of 101,526 segments. No further preprocessing of the raw data was carried out.

The relative distribution of arrhythmia classes labeled by the expert cardiologist is shown in figure 2, and indicates an imbalanced dataset, with a ratio of 0.5% (Fusion of Ventricular and Normal as the least represented class) to 88% (Normal Beat as the dominant class). A detailed account of the number of classes present in each recording has been provided in the Appendix.

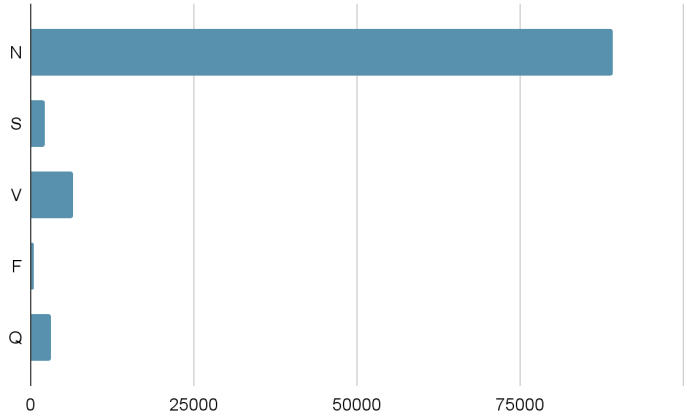


Figure 2. Distribution of heartbeat classes in the dataset.

3. METHODOLOGY

3.1. Neural Network Arrhythmia Classifier

A 1D convolutional neural network was trained for the task of arrhythmia classification. The 1D convolutional layer input array is component-by-component multiplied by the kernel then the resulting products are summed as demonstrated in the following equation.

$$f(i) = \Sigma_{n=1}^{v\ k} S(i + n)K(n)$$

Additionally to the sum of these values is added a bias value then fed to an activation function to find the output value. This process is repeated along the input temporal axis until the entirety of the input array is processed. The input array width shares the same value as the kernels. Thus kernel values are manually assigned and composed of number, length, and sliding window size. During the training process, the network optimizer continuously updates the kernel weights. Convolutional layers are composed of attributes that enhance machine learning systems namely: parameter sharing, equivalent representations, and sparse weights [27]. The aforementioned

attributes lead to improvements in the statistical efficacy of automated feature extraction and learning of local features residing within the data. Convolution layers have 32 kernels of size 5.

The CNN model, illustrated in figure 3, employs connections in a similar manner to Residual Connections, introduced by He K *et al*. [28]. Each convolution layer is followed by 5 residual blocks composed of 2 convolution layers, 2 ReLU (Rectified Linear Unit) activations, and a max-pooling of stride 2 and size 5 in all pooling layers, the operation that extracts the maximum value output within the specified size-shifting along the direction of the time-series. Followed by one fully connected (FC) layer containing 32 neurons. The fully connected layer is interconnected to all units in the forward layer. FC layer is calculated as follows:

$$y = f(\Sigma u \times w + b)$$

The input u is multiplied by the weights w then products are totaled, and bias b is added. The activation function f receives as input the result, which will calculate the output. The biases b and weights w are trainable variables, in contrast, the activation function is manually selected (ReLU) and calculated as follows:

$$y = max(0, u)$$

To predict the output class probabilities, a softmax layer is applied to output structural-state identification results. The probabilities of all predictive prospects are measured, thus the final result represents the one with the highest possibility. Calculated using the following equation:

$$y_i = \frac{exp(u_i)}{\Sigma_{i=1}^{n} exp(u_i)}.$$

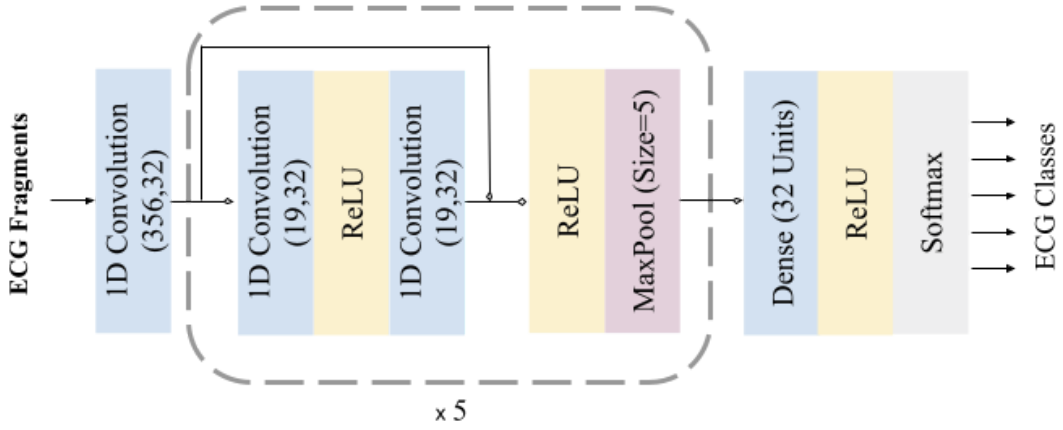


Figure 3. The 1D deep convolutional neural network architecture, adopted for the arrhythmia classification task. Resulting in a 12-layer model.

3.2. Implementation and Training

It is evident from figure 2 that the dataset is fairly imbalanced with unequal distribution of different arrhythmia classes. To prevent potential biases towards more dominant classes, the approach of random over-sampling examples (ROSE) was adopted [29], which augments the data after removing the baseline. This resulted in a balanced dataset, with 76,607 samples for each arrhythmia class. The dataset was then randomly split into training (287,276 samples) and validation (95,759 samples), with testing samples selective split prior to over-sampling (3500 samples).

The network was implemented using the Tensorflow library [30]. Our source code is available Online [31]. The loss function used was cross-entropy on the softmax predicted outputs. Adam

optimizer [32] was applied with decaying learning rates reducing at a factor of 0.75 for every ten thousand iterations. Training the entire network takes less than 15 minutes with a mini-batch size of 256 samples over 100 epochs using an Nvidia Tesla P100 processor. The validation dataset was used for early stopping to avoid redundant training and overfitting, with patience of 10 epochs. The model was trained until the validation loss plateaued.

3.3. Network optimization for edge computing

Edge and wearable devices have relatively much smaller, low-power, and slower processors, compared to desktop processors. Therefore, for the continuous execution of the arrhythmia classifier on such devices with limited memory and computational power and in order to meet the timing requirements, further optimization of the model was applied to reduce its size (smaller storage size and less memory usage) and latency, while maintaining (or with little degradation in) the model accuracy.

To this end, post-training full integer quantization was adopted and applied to the CNN model developed using Tensorflow Lite deep learning library [33]. The full integer (8-bit) quantization technique [34], approximates floating-point values in the trained model, layer by layer, as

$$real\_value = (int8\_value - zero\_point) \times scale$$

Per-channel weights are represented by int8 two's complement values in the range [-127, 127] with zero-point equal to 0. Per-tensor activations/inputs are represented by int8 two's complement values in the range [-128, 127], with a zero point in the same range [35]. In this

process, only floating-point weights are quantized to 8-bit integer precision in a bit-by-bit operation [35], in an iterative process until the network is fully mapped.

As a proof of concept, two hardware platforms were used as examples of edge low-powered processors for running the quantized neural network: ARM Cortex A53 and ARM Cortex A55. Further details for the platforms are provided in Table 5.

Device	Raspberry Pi v2 	Huawei P30 Pro 
Chipset	Broadcom BCM2836	Kirin 980
CPU	ARM Cortex A53	ARM Cortex A55/A76
Mean Inference Time	4.76 ms	7.65 ms

Table 2. Two hardware platforms are used as low-powered edge processing devices.

3.4. Real-time Continuous Monitoring

In order to investigate whether our proposed approach can be employed for real-time applications, a continuous monitoring scenario was simulated where previously acquired ECG signals were transmitted as segments of contiguous time windows over Bluetooth from a laptop. The duration of each time window was 1s, and one segment was transmitted every second to

mimic a real-time acquisition from the patient. Transmitted data was received by the mobile phone device running the quantized classifier (figure 4), updating the predicted arrhythmia class every second.

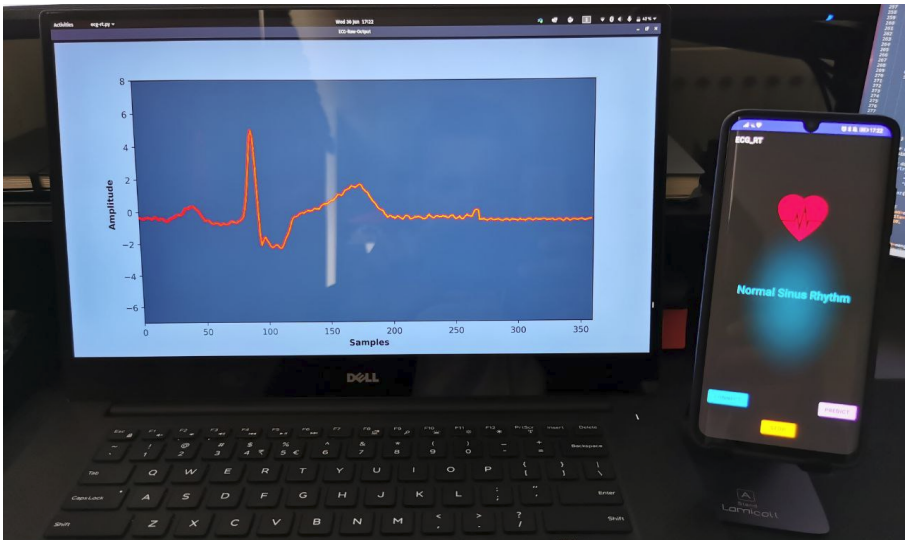


Figure 4. Simulated real-time continuous monitoring scenario: a laptop transmitting previously recorded ECG signal over Bluetooth to a mobile phone which is predicting an arrhythmia class every second.

3.5. Evaluation Metrics

The classification was evaluated using the standard measures: classification accuracy (Acc), sensitivity (Sen), specificity (Spe), F_1 score, and confusion matrix.

Besides accuracy, execution time (computational intensity) and power consumption of the arrhythmia classifier were also measured as the other two important factors, particularly for the application of continuous monitoring on wearable devices.

The inference time was measured by recording the average time in milliseconds for all predictions made on the test dataset. In order to be able to measure the battery usage by the smartphone application, the fully charged phone was left idle in flight mode for 12 hours, and the dropped battery percentage was recorded at the end of this period. The battery was then fully charged again, and the application was kept continuously running for the same period of time, and the battery percentage was recorded. The difference in the two percentages was assumed to be the quicker battery drainage due to the application. The energy $E_{(\mathrm{Wh})}$ in Watt-hours was then estimated as

$$E_{(\mathrm{Wh})} = Q_{(\mathrm{mAh})} \times V_{(\mathrm{V})} / 1000$$

where $Q_{(\mathrm{mAh})}$ is equal to the electric charge (capacity of the battery) in milliamp-hours, and $V_{(\mathrm{V})}$ is the voltage in volts. This process was repeated several times and average power consumption was calculated.

Additionally, the energy consumption *E* per prediction (i.e., classifying each ECG segment) was estimated in Joules as

$$E_{(\mathrm{J})} = P_{(\mathrm{W})} \times t_{(\mathrm{s})}$$

where *P* is the nominal power rating of the ARM Cortex A55 in Watts, and *t* is average inference time per prediction [36].

4. RESULTS AND DISCUSSION

This section provides and discusses the results obtained at each stage of the development described previously.

4.1. Arrhythmia Classification

The mean inference time per classification (i.e., for each ECG segment) was ~30 ms using an Nvidia Tesla P100 processor and ~16 ms using an Nvidia RTX 3090 processor.

Table 3 and figure 5 provided the prediction results for the arrhythmia classifier (non-quantized) when applied to the test dataset. It shows that the model was able to correlate the characteristics with the correspondent arrhythmia. The model detected all of the anomalous segments for Ventricular Ectopic Beat (V class) reliably. Only for one anomaly, Fusion of Ventricular and Normal (F class), the detection was relatively less reliable with a sensitivity of 99.70%, probably because of the F class being the least represented class in the dataset (0.5% of the dataset). Overall, 99.88% of anomalous segments in the test set were detected. Of the normal heartbeats, 0.25% were falsely indicated as being anomalous.

Class	Original neural network				Quantized neural network			
	Acc	*Sen*	*Spe*	F_1	*Acc*	*Sen*	*Spe*	F_1
N	99.9%	99.8%	99.9%	99.8%	99.4%	99.8%	99.3%	98.6%
S	99.9%	99.9%	99.9%	99.8%	99.7%	99.1%	99.9%	99.4%
V	99.9%	100%	99.9%	99.9%	99.6%	99.5%	99.7%	99.3%
F	99.9%	99.7%	100%	99.8%	99.5%	99.4%	99.9%	96.7%
Q	99.9%	99.9%	100%	99.9%	99.9%	99.9%	100%	99.9%

Table 3. Results of arrhythmia classification for the test dataset.

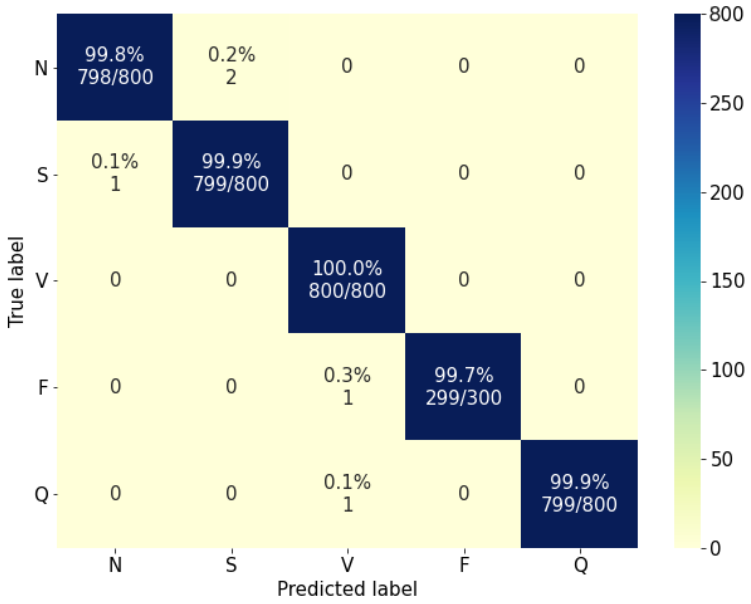


Figure 5. Original neural network confusion matrix.

In order to visualize the proximity between the feature vectors, the t-distributed stochastic neighbor embedding (t-SNE) is plotted in figure 6, which displays a planar representation of the internal high-dimensional organization of the 5 ECG classes within the network's final hidden layer (i.e., input data of the fully connected layer). As can be seen, the colors, representing the extracted features of different types of ECG labels, clearly separated into five clusters with minor overlap, indicating that the features learned by the model are discriminative for classifying different arrhythmia classes. The most pronounced overlapping is in distinguishing F from V signals (0.3% misclassified).

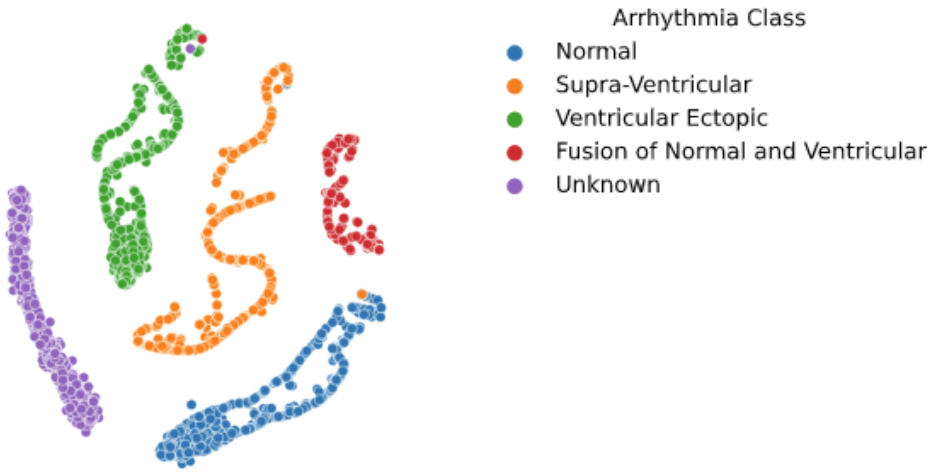


Figure 6. t-Distributed Stochastic Neighbor Embedding (t-SNE) visualization for five AAMI classes. Each dot represents an ECG signal sample from the test subset extracted from the MIT-BIH arrhythmia database for each category; different colors represent different arrhythmia classes (best viewed in color).

4.2. Network Performance on Edge Devices

The quantized classifier showed a small degradation in performance as shown in Table 3; a minute drop in accuracy (≤0.3%) across all class types; sensitivity and specificity were above 99.1% and 99.3% for all anomaly types, respectively. Figure 7 displays a relative confusion matrix demonstrating the difference between the two (original and quantized) classifiers.

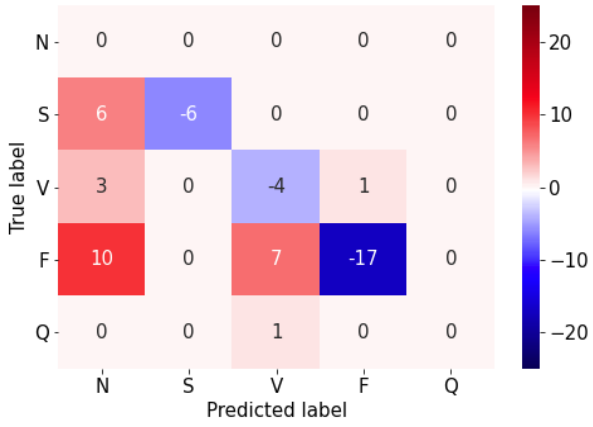


Figure 7. Relative confusion matrix displaying the difference in predictions between the original (non-quantized) and quantized classifiers.

As shown in figure 9, the quantized model however achieved significant improvements in inference (prediction) time, with a reduced inference time per classification of 12.13±0.61 (11.99, 31.14) and 7.69±0.58 (7.00, 24.00) and 4.76±0.04 (4.70, 12.17) milliseconds on the GPU, smartphone, and Raspberry Pi processors, respectively. The reduction in the model size was also noticeable; the size of the quantized model was 93 kB, compared to 853 kB for the original model.

4.3. Performance Evaluations in Presence of Noise

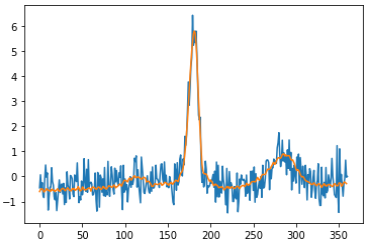
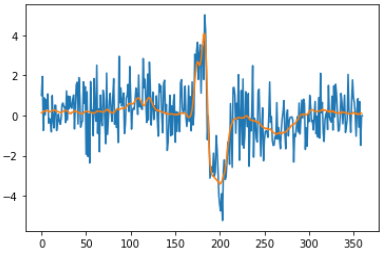
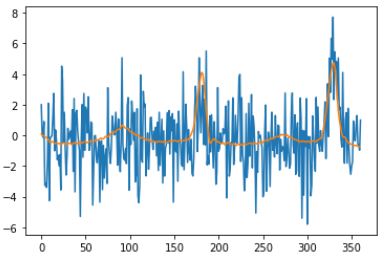
Since the MIT-BIH arrhythmia database contains solely signals recorded from an ambulatory device in a controlled environment, it may not represent all the artifacts present in the ECG data acquired from mobile devices. Therefore, in order to demonstrate the feasibility of our approach in a wearable deployment scenario, synthetic noise was added to the ambulatory ECG recordings.

Contaminated signals were generated replacing baseline ambulatory ECG signals with a matching signal-to-noise ratio for low amplitude noise, and high-frequency noise overlapping PQRST complex for artifacts such as motion and muscle contraction.

Four different types of noise and artifacts consistent with mobile devices [37], depicted in Table 4, were considered. A band-pass filter was applied to attenuate signal frequencies, anti-aliasing, and limit saturation [38], equating to similar signal morphologies, apart from edge cases due to external signal interference such as noise/artifacts.

Since these artifacts (e.g., motion) are not always present, three scenarios were considered in which different percentages of the entire dataset were randomly selected and contaminated by the noise; 20%, 50%, and 100% indicating that noise is always present. Care was taken to ensure the same percentage of noisy signals was present in both training and testing datasets.

The classifier was then trained and tested using the noisy data using the same methodology detailed in section 3.

Noise / Artifact	Signal Sample
Electrode contact noise	
Power line interference	
Instrumentation noise	

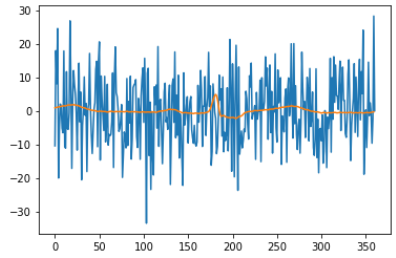
Motion / Muscle contraction	

Table 4. Examples of the synthetic noise added to the ambulatory ECG recordings to mimic the artifacts in mobile devices; the added noise is overlaid on the original ECG signal.

An expected degradation in the performance of the classifier can be observed when exposed to noisy data. Figure 8 demonstrates a relative confusion matrix for the classifier when exposed to the noisy data with 20% corrupted signals when compared to baseline ambulatory ECG recordings. A drop of 3.3% in the average accuracy was observed. However, the sensitivity remained above 90.1% for categories N, V, Q, and over 78.3% for S and F; and the specificity was still above 98.4% for all anomaly types. When half the signals are corrupted (i.e., 50% noisy), the average accuracy degrades to 7.7%. With sensitivity above 84.6% for categories N and Q, 76.5% for V, and above 61.7% for S and F; specificity above 99.4% for all anomaly types. Finally, when exposed to 100% noisy data, the average accuracy drops by 12.9%. With sensitivity for each category of 99.9% (N), 43.1% (S), 62.9% (V), 32.6% (F), and 71.9% (Q); and specificity above 98.4% for all anomaly types.

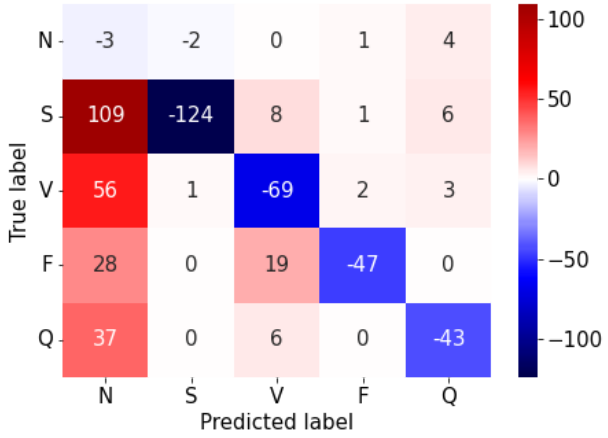


Figure 8. The relative confusion matrix indicates a small deterioration in the performance of the classifier when exposed to the noisy data (20%) when compared to baseline ambulatory ECG recordings.

4.4. Continuous Arrhythmia Monitoring

The home screen of the Android smartphone application is shown in figure 10. The application has a simple and intuitive interface; the user must click on the CONNECT button first, which will instruct the phone to look for and connect to the device transmitting the ECG signal (here, the laptop) via Bluetooth. With the connection established, the application is ready for continuous classification, which will be started after clicking the PREDICT button. The predicted arrhythmia class by the quantized model is displayed in the middle of the screen. This is updated in real-time as soon as the next ECG segment has arrived and been analyzed.

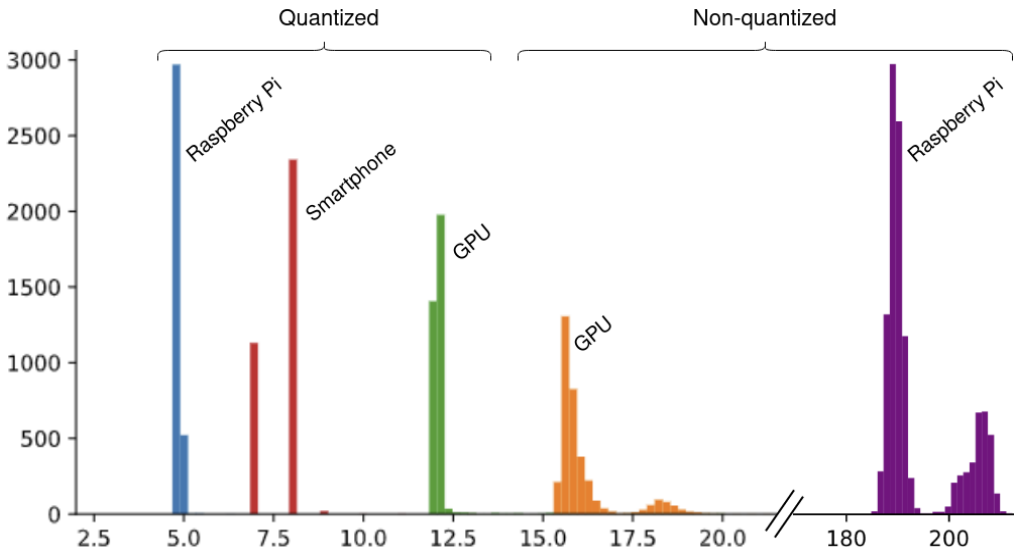


Figure 9. Inference time distributions per prediction for GPU Nvidia RTX3090, Raspberry Pi (ARM Cortex A53), and smartphone (ARM Cortex A55) for the original (non-quantized) and quantized neural networks: non-quantized GPU, 16.26±3.92ms; quantized GPU, 12.13±0.61ms; non-quantized A53, 198.75±59.11ms; quantized A53, 4.76±0.04ms; quantized smartphone 7.65±0.58ms.

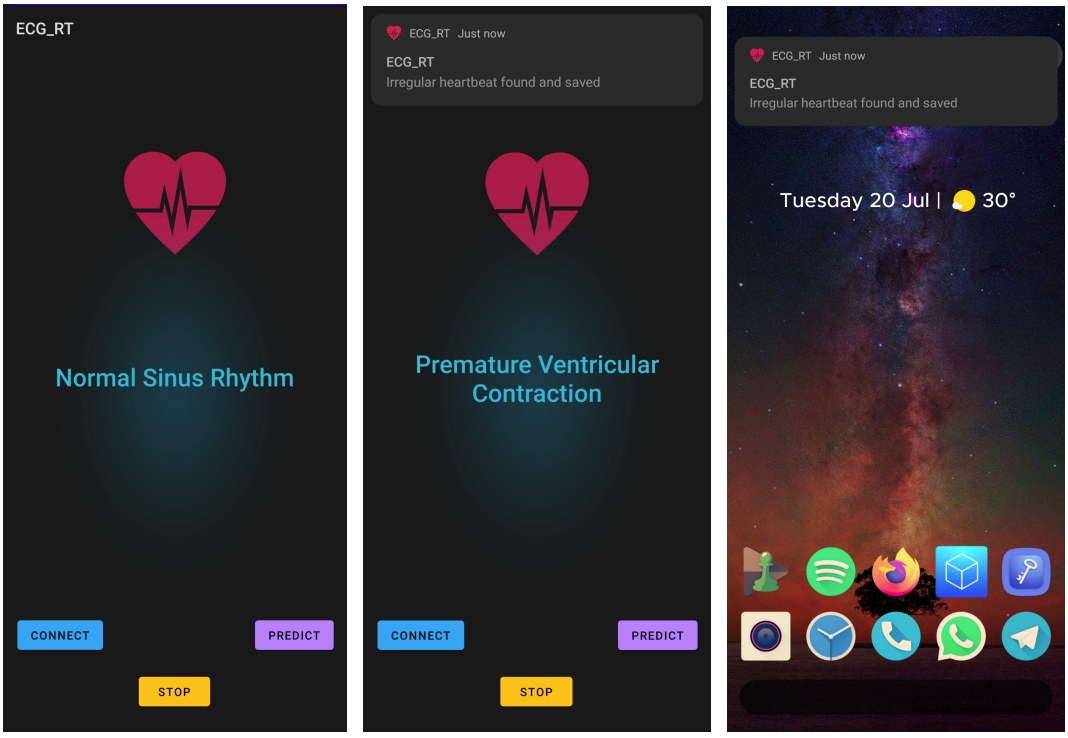


Figure 10. Smartphone application running the quantized model for the continuous ECG arrhythmia detection. Left: normal sinus rhythm predicted and displayed on the home screen. Middle: irregular heartbeat detected and displayed, also triggering a pop-up window. Right: the pop-up window when the application is running in the background.

When an abnormal arrhythmia class is predicted, a message is shown on the home screen and the predicted class together with the corresponding ECG signal and timestamp will be saved for further scrutiny (figure 10, middle panel). The application can also perform while not in the foreground, in which case the user is notified by triggering a pop-up window when an anomaly is detected (figure 10, right panel). This process is repeated indefinitely or until the user chooses to stop the execution of the application.

The smartphone activity interacting with the application over a period of 1s is displayed in figure 11, providing a breakdown of the application's CPU usage. The prediction plus producing the output display is carried out in less than 300ms, leaving the application idle >70% of the time for the next segment of the ECG signal to arrive. This clearly demonstrates the time constraint for a real-time application is satisfied.

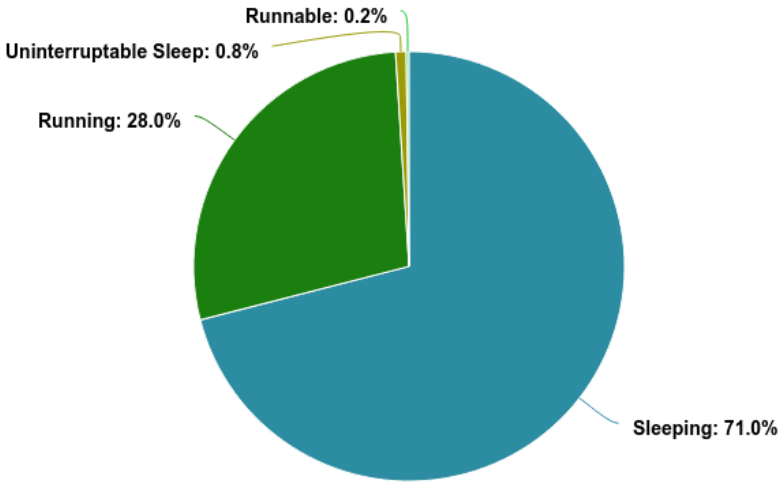


Figure 11. The average percentage of total execution time for each ECG segment on the smartphone application. Data collected using Android OS Systrace tool [39], logging system changes, running over 1000 ms. *Sleeping* represents the application's idle time (710 ms). *Uninterruptible Sleep* is the system-locked block that cannot be interrupted (8 ms). *Runnable* indicates the application scheduler's intent to run the designated thread (2 ms). *Running* is the representation of the application's active execution time (280 ms) when running inference and displaying results.

The battery usage for the arrhythmia classification application using the quantized classifier was 2.04Wh for 12 hours of continuous running. Considering only the nominal power rating of 0.765W for the smartphone CPU used in this study and the average inference time of 7.65ms, the energy consumption per prediction was 5.85mJ.

The energy consumption for the Raspberry Pi CPU with a nominal power rating of 0.9W and average inference time of 4.76ms was 4.28mJ, compared with 178.88mJ for the non-quantized classifier with an average inference time of 198.75ms.

4.5. Comparison With Existing Work

Table 5 compares the proposed lightweight solution with the existing work, in terms of classification performance, and when the 5 AAMI classes were of interest; studies with binary (normal vs abnormal) classification were excluded. The proposed classifier has comparable classification performance with respect to the existing counterparts.

Work	*Acc*	*Sen*	*Spe*	*Ppr*	F_1	Type
Kachuee *et al.* [40]	93.4%	-	-	-	-	CNN
Acharya *et al.* [8]	93.5%	96.0%	91.6%	97.87	-	CNN
Martis *et al.* [41]	93.8%	99.5%	97.4%	99.1%	-	DWT + SVM
Li *et al.* [42]	94.6%	-	-	-	-	DWT + Random Forest
Martis *et al.* [43]	99.3%	99.9%	99.8%	99.2%	-	Probabilistic NN
Proposed	**99.6%**	**98.5%**	**99.8%**	**99.2%**	**98.8%**	**CNN**

Proposed / 20% noise	**96.4%**	**89.3%**	**97.6%**	**94.2%**	**90.6%**	**CNN**
Proposed / 50% noise	**91.9%**	**77.5%**	**94.8%**	**88.8%**	**80.2%**	**CNN**
Proposed / 100% noise	**86.7%**	**62.1%**	**91.4%**	**85.5%**	**65.1%**	**CNN**

Table 5. Classification performance comparison between the proposed lightweight solution and previous works using 5 AAMI classes.

5. CONCLUSION

In this study we proposed a deep convolutional neural network for automated classification of five different categories of arrhythmia. In all anomalous cases, the classification accuracy achieved an F_1 score of ≥99.8%.

The proposed model uses raw ECG signals, without the need for any preprocessing steps of noise removal or beat detection. This facilitated the optimization of the classifier, which was achieved by quantizing the neural network using a post-training 8-bit integer quantization technique. This resulted in a lightweight model with a smaller footprint in terms of memory and latency, and with small degradation classification accuracy, in relation to its non-quantized counterpart.

The employed arrhythmia database contains solely signals recorded from an ambulatory device in a controlled environment. In order to represent artifacts consistent with the wearable devices, different types of synthetic noise were added to the ambulatory ECG recordings. However, in our follow-up studies, we will use signals acquired from mobile devices.

To demonstrate the feasibility of real-time 24h continuous arrhythmia monitoring, we produced a working prototype using an ordinary smartphone as an edge processing device, with 7.65ms inference time and 5.85mJ energy consumption, per prediction.

The availability of such low-cost systems for continuous ambulatory cardiac monitoring would provide privacy, energy efficiency, low latency, low storage usage, and portability, empowering clinicians in the accurate diagnosis of early cardiac diseases, and constant monitoring of patients. Using such wearable devices on a continuous basis will potentially be highly beneficial in early detection and prevention of medical complications and emergencies, particularly in cardiac patients and among elderly populations [43].

Acknowledgment

Funding: Massoud Zolgharni is supported by the British Heart Foundation, United Kingdom (Grant no. PG/19/78/34733). Henrique De Melo Ribeiro is supported by the Vice Chancellor's Scholarship at the University of West London.

REFERENCES

[1] World Health Organization. WHO | Cardiovascular diseases (CVDs) Available at: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds), 2017 (Accessed 19 May 2021).

[2] Schlant, R.C., Adolph, R.J., DiMarco, J.P., Dreifus, L.S., Dunn, M.I., Fisch, C., Garson Jr, A., Haywood, L.J., Levine, H.J. and Murray, J.A., 1992. Guidelines for electrocardiography. A report of the American College of Cardiology/American Heart Association Task Force on Assessment of Diagnostic and Therapeutic Cardiovascular Procedures (Committee on Electrocardiography). Circulation, 85(3), pp.1221-1228.

[3] Kuehn, B.M., 2016. Telemedicine helps cardiologists extend their reach. Circulation, 134(16), pp.1189-1191.

[4] Saadatnejad, S., Oveisi, M. and Hashemi, M., 2019. LSTM-based ECG classification for continuous monitoring on personal wearable devices. IEEE journal of biomedical and health informatics, 24(2), pp.515-523.

[5] Li, Y., Pang, Y., Wang, J. and Li, X., 2018. Patient-specific ECG classification by deeper CNN from generic to dedicated. Neurocomputing, 314, pp.336-346.

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

[7] Acharya, U.R., Fujita, H., Lih, O.S., Hagiwara, Y., Tan, J.H. and Adam, M., 2017. Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network. Information sciences, 405, pp.81-90.

[8] Acharya, U.R., Oh, S.L., Hagiwara, Y., Tan, J.H., Adam, M., Gertych, A. and San Tan, R., 2017. A deep convolutional neural network model to classify heartbeats. Computers in biology and medicine, 89, pp.389-396.

[9] Yıldırım, Ö., Pławiak, P., Tan, R.S. and Acharya, U.R., 2018. Arrhythmia detection using deep convolutional neural network with long duration ECG signals. Computers in biology and medicine, 102, pp.411-420.

[10] Zhou, S. and Tan, B., 2020. Electrocardiogram soft computing using hybrid deep learning CNN-ELM. applied soft computing, 86, p.105778.

[11] Chan, C.C., Chen, C.W., Chou, W.C., Ho, Y.L., Lin, Y.H. and Ma, H.P., 2012, November. Live demonstration: A mobile ECG healthcare platform. In 2012 IEEE Biomedical Circuits and Systems Conference (BioCAS) (pp. 87-87). IEEE.

[12] Gakare, P.K., Patel, A.M., Vaghela, J.R. and Awale, R.N., 2012, October. Real time feature extraction of ECG signal on android platform. In 2012 international conference on communication, information & computing technology (ICCICT) (pp. 1-5). IEEE.

[13] Guo, X., Duan, X., Gao, H., Huang, A. and Jiao, B., 2013. An ecg monitoring and alarming system based on android smart phone. Communications and Network, 5(03), p.584.

[14] Singh, P. and Jasuja, A., 2017, May. IoT based low-cost distant patient ECG monitoring system. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 1330-1334). IEEE.

[15] Xia, Y., Zhang, H., Xu, L., Gao, Z., Zhang, H., Liu, H. and Li, S., 2018. An automatic cardiac arrhythmia classification system with wearable electrocardiogram. IEEE Access, 6, pp.16529-16538.

[16] Xia, Y. and Xie, Y., 2019. A novel wearable electrocardiogram classification system using convolutional neural networks and active learning. IEEE Access, 7, pp.7989-8001.

[17] Wang, N., Zhou, J., Dai, G., Huang, J. and Xie, Y., 2019. Energy-efficient intelligent ECG monitoring for wearable devices. IEEE transactions on biomedical circuits and systems, 13(5), pp.1112-1121.

[18] Saadatnejad, S., Oveisi, M. and Hashemi, M., 2019. LSTM-based ECG classification for continuous monitoring on personal wearable devices. IEEE journal of biomedical and health informatics, 24(2), pp.515-523.

[19] Yan, Z., Zhou, J. and Wong, W.F., 2021. Energy efficient ECG classification with spiking neural network. Biomedical Signal Processing and Control, 63, p.102170.

[20] Bauer, F.C., Muir, D.R. and Indiveri, G., 2019. Real-time ultra-low power ECG anomaly detection using an event-driven neuromorphic processor. IEEE transactions on biomedical circuits and systems, 13(6), pp.1575-1582.

[21] Alickovic, E. and Subasi, A., 2015. Effect of multiscale PCA de-noising in ECG beat classification for diagnosis of cardiovascular diseases. Circuits, Systems, and Signal Processing, 34(2), pp.513-533.

[22] Lin, C.C. and Yang, C.M., 2014. Heartbeat classification using normalized RR intervals and morphological features. Mathematical Problems in Engineering, 2014.

[23] Mar, T., Zaunseder, S., Martínez, J.P., Llamedo, M. and Poll, R., 2011. Optimization of ECG classification by means of feature selection. IEEE transactions on Biomedical Engineering, 58(8), pp.2168-2177.

[24] Goldberger, A.L., Amaral, L.A., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K. and Stanley, H.E., 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. circulation, 101(23), pp.e215-e220.

[25] A. for the Advancement of Medical Instrumentation et al., 1998. Testing and reporting performance results of cardiac rhythm and st segment measurement algorithms, ANSI/AAMI EC38, vol. 1998.

[26] De Chazal, P., O'Dwyer, M. and Reilly, R.B., 2004. Automatic classification of heartbeats using ECG morphology and heartbeat interval features. IEEE transactions on biomedical engineering, 51(7), pp.1196-1206.

[27] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), pp.436-444.

[28] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[29] Menardi, G. and Torelli, N., 2014. Training and assessing classification rules with imbalanced data. Data mining and knowledge discovery, 28(1), pp.92-122.

[30] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M. and Ghemawat, S., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

[31] Intelligent Sensing & Vision. 2021. ECG-based Arrhythmia Monitoring using Quantized Neural Networks. [ONLINE] Available at: https://intsav.github.io/realtime_ecg.html. (Accessed 15 July 2021).

[32] Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[33] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H. and Kalenichenko, D., 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2704-2713).

[34] Wu, J., Leng, C., Wang, Y., Hu, Q. and Cheng, J., 2016. Quantized convolutional neural networks for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4820-4828).

[35] Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R. and Bengio, Y., 2017. Quantized neural networks: Training neural networks with low precision weights and activations. The Journal of Machine Learning Research, 18(1), pp.6869-6898.

[36] Amirshahi, A. and Hashemi, M., 2019. ECG classification algorithm based on STDP and R-STDP neural networks for real-time monitoring on ultra low-power personal wearable devices. IEEE transactions on biomedical circuits and systems, 13(6), pp.1483-1493.

[37]Rosu, M.C. and Hamed, Y., 2015, June. Methods for denoising the ECG signal in wearable systems. In 2015 7th International Conference on Electronics, Computers and Artificial Intelligence (ECAI) (pp. 1-6). IEEE.

[38] Mark, R.G., Schluter, P.S., Moody, G., Devlin, P. and Chernoff, D., 1982, January. An annotated ECG database for evaluating arrhythmia detectors. In IEEE Transactions on Biomedical Engineering (Vol. 29, No. 8, pp. 600-600). 345 E 47TH ST, NEW YORK, NY 10017-2394: IEEE-INST ELECTRICAL ELECTRONICS ENGINEERS INC.

[39] Perfetto. 2021. Quickstart: Record traces on Android - Perfetto Tracing Docs. [ONLINE] Available at: https://perfetto.dev/docs/quickstart/android-tracing. (Accessed 21 June 2021).

[40] Kachuee, M., Fazeli, S. and Sarrafzadeh, M., 2018, June. Ecg heartbeat classification: A deep transferable representation. In 2018 IEEE International Conference on Healthcare Informatics (ICHI) (pp. 443-444). IEEE.

[41] Martis, R.J., Acharya, U.R., Lim, C.M., Mandana, K.M., Ray, A.K. and Chakraborty, C., 2013. Application of higher order cumulant features for cardiac health diagnosis using ECG signals. International journal of neural systems, 23(04), p.1350014.

[42] Li, T. and Zhou, M., 2016. ECG classification using wavelet packet entropy and random forests. Entropy, 18(8), p.285.

[43] Martis, R.J., Acharya, U.R. and Min, L.C., 2013. ECG beat classification using PCA, LDA, ICA and discrete wavelet transform. Biomedical Signal Processing and Control, 8(5), pp.437-448.

[44] Kekade, S., Hseieh, C.H., Islam, M.M., Atique, S., Khalfan, A.M., Li, Y.C. and Abdul, S.S., 2018. The usefulness and actual use of wearable devices among the elderly population. Computer methods and programs in biomedicine, 153, pp.137-159.

Appendix

Number of heartbeats belonging to each class present in each recording, as determined by the expert cardiologists:

	AAMI category

Record id	N	S	V	F	Q
100	2237	33	1	0	0
101	1859	3	0	0	2
103	2081	2	0	0	0
105	2526	0	41	0	5
106	1507	0	520	0	0
107	2077	0	59	0	0
108	1738	5	17	2	0
109	2490	0	38	2	0
111	2123	0	1	0	0
112	2535	2	0	0	0
113	1787	6	0	0	0
114	1820	12	43	4	0
115	1951	0	0	0	0
116	2301	1	109	0	0
117	1533	1	0	0	0
118	2165	96	16	0	0
119	1543	0	444	0	0
121	1859	1	1	0	0
122	2474	0	0	0	0
123	1514	0	3	0	0
124	1530	36	47	5	0
200	1742	30	826	2	0
201	1624	138	198	2	0
202	2060	55	19	1	0
203	2528	2	444	1	4
205	2570	3	71	11	0
207	1542	107	210	0	0
208	1585	2	992	372	2

209	2620	383	1	0	0
210	2421	22	195	10	0
212	2747	0	0	0	0
213	2639	28	220	362	0
214	2001	0	256	1	2
215	3193	3	164	1	0
217	244	0	162	0	1802
219	2082	7	64	1	0
220	1952	94	0	0	0
221	2031	0	396	0	0
222	2060	421	0	0	0
223	2028	89	473	14	0
228	1687	3	362	0	0
230	2254	0	1	0	0
231	1567	1	2	0	0
232	397	1383	0	0	0
233	2229	7	830	11	0
234	2699	50	3	0	0

FIGURES

Figure 1.

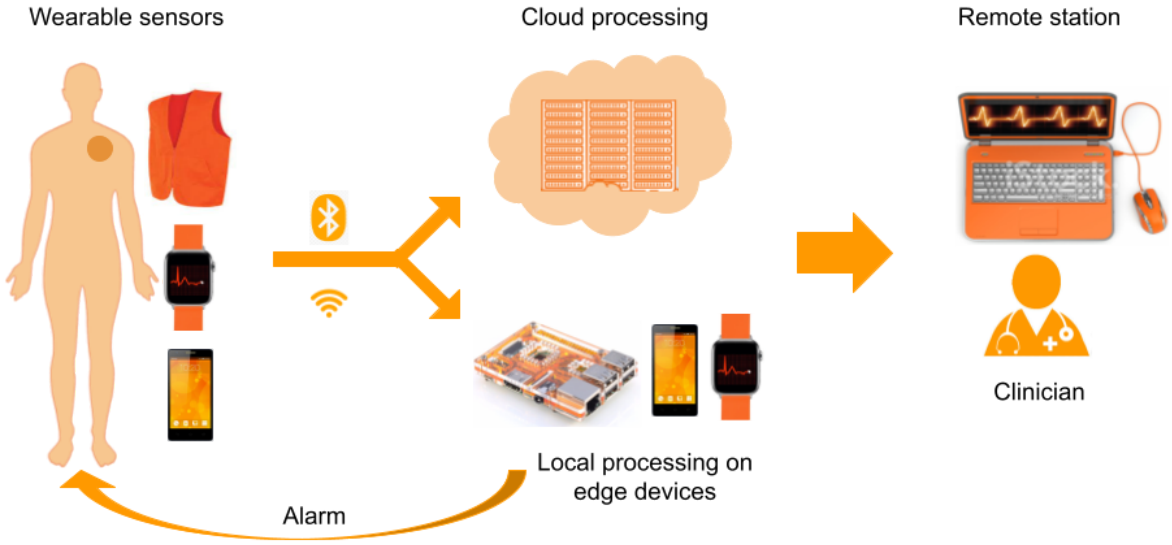


Figure 2.

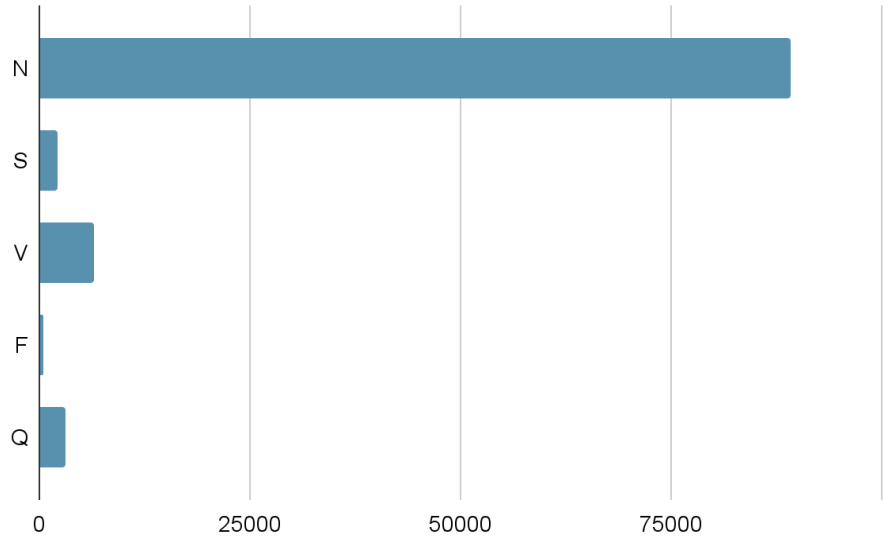


Figure 3.

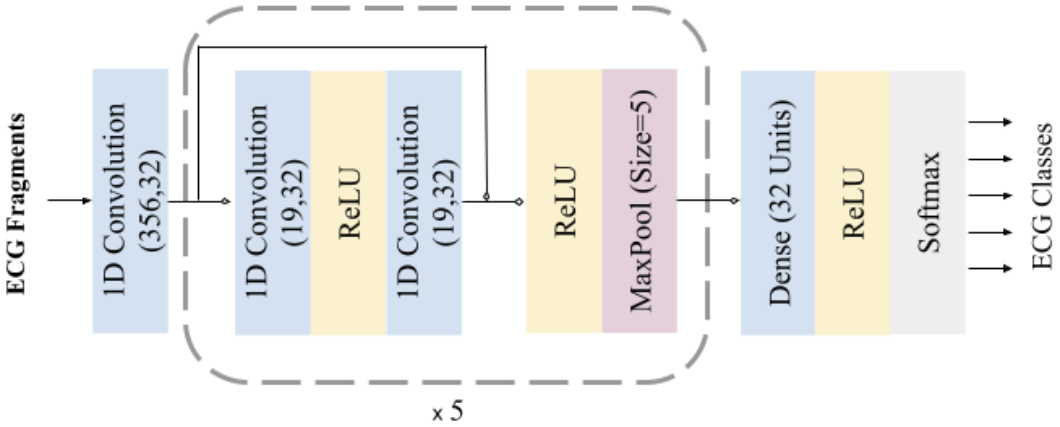


Figure 4.

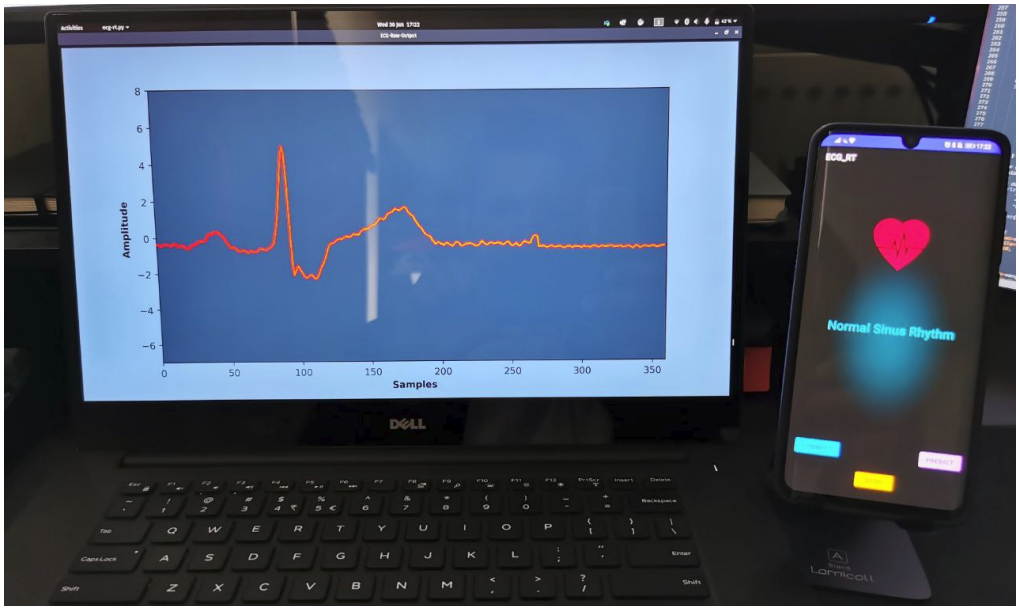


Figure 5.

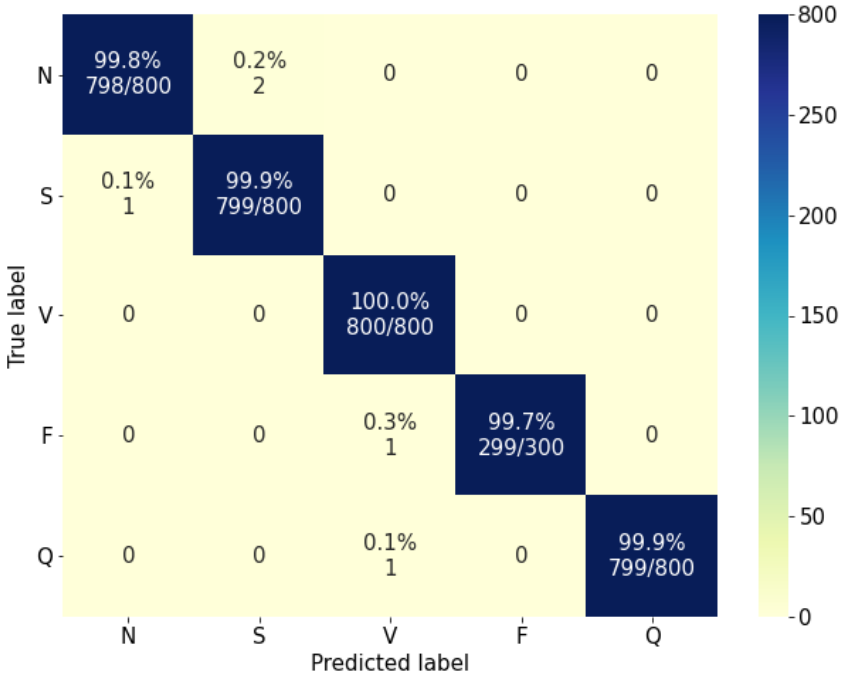


Figure 6.

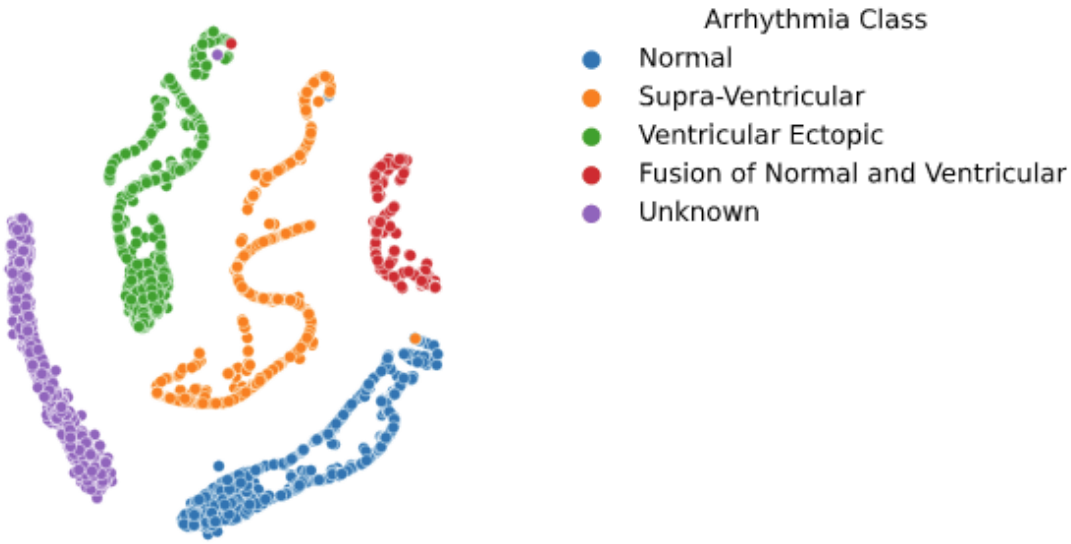


Figure 7.

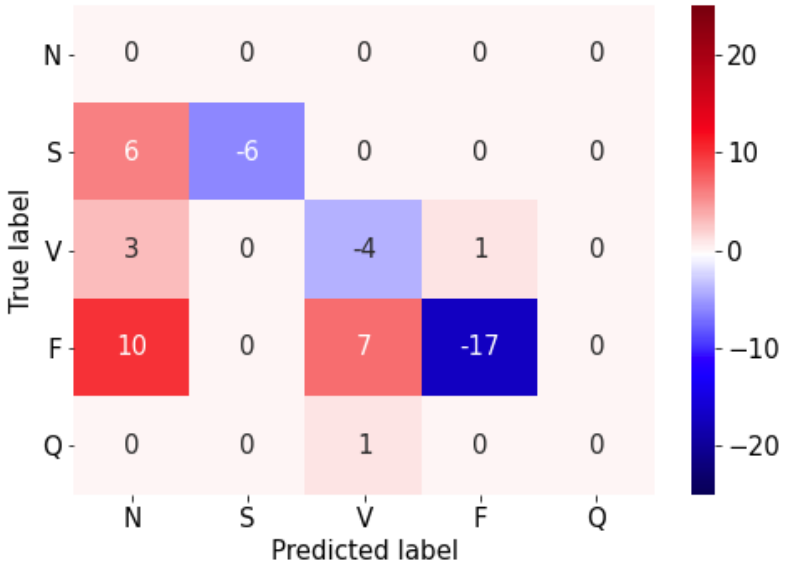


Figure 8.

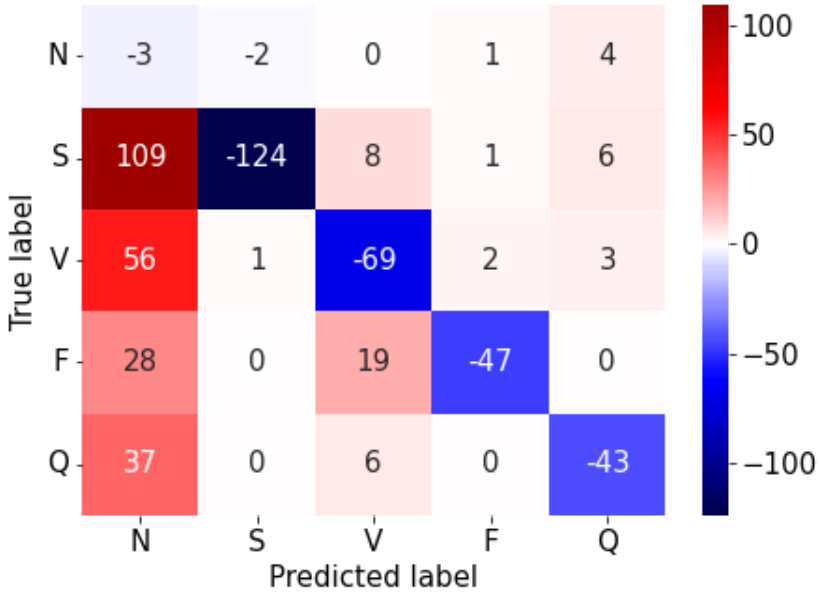


Figure 9.

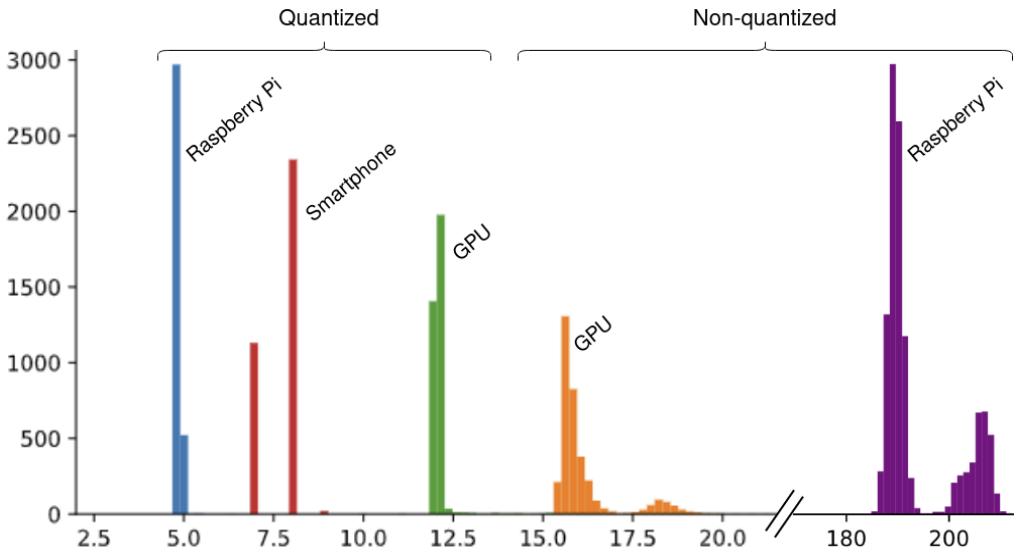


Figure 10.

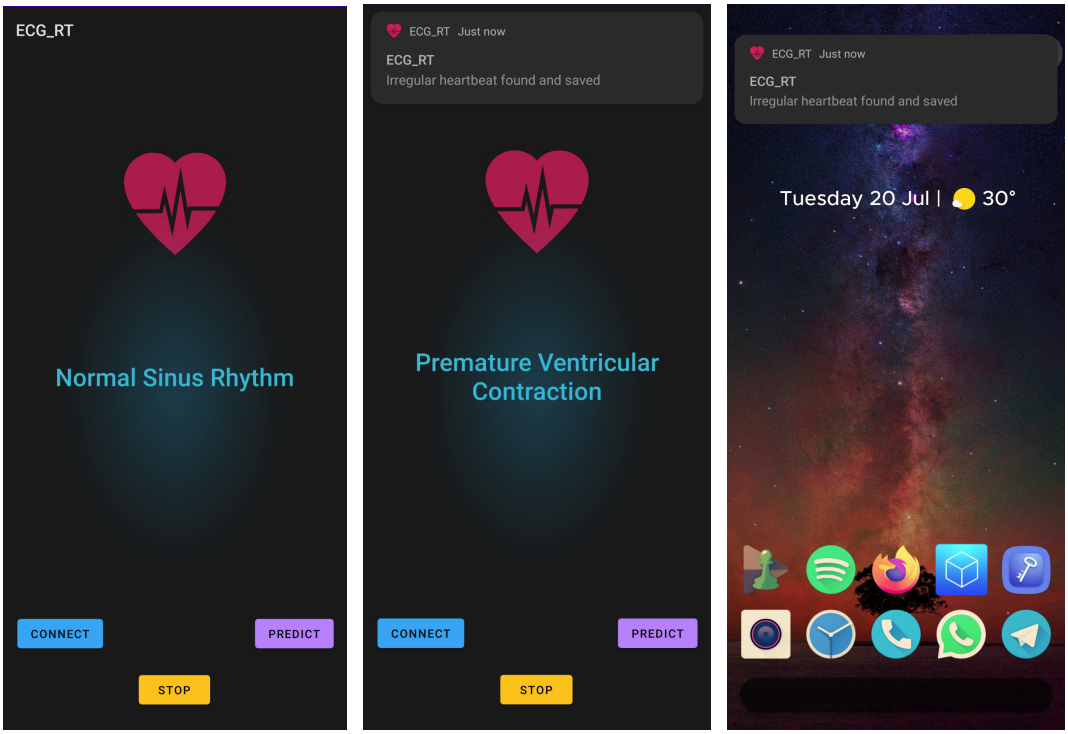
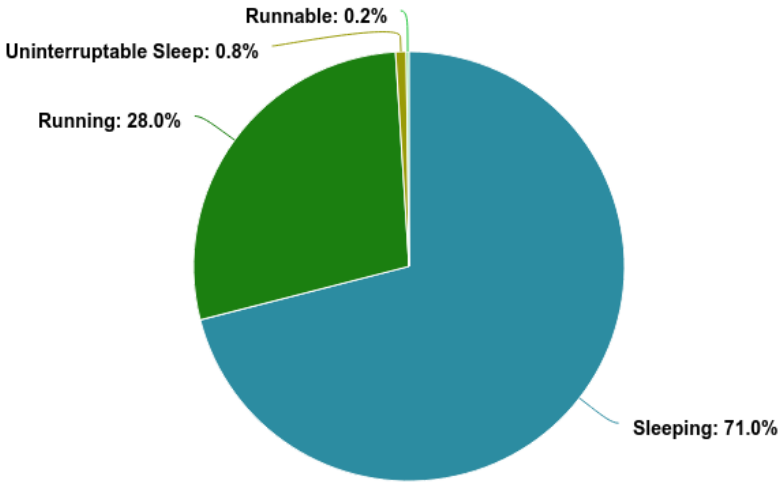


Figure 11.


Runnable: 0.2%
Uninterruptable Sleep: 0.8%
Running: 28.0%
Sleeping: 71.0%

TABLES

Table 1.

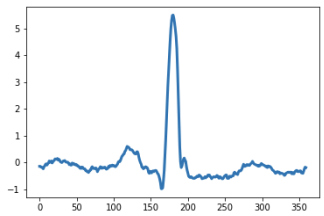
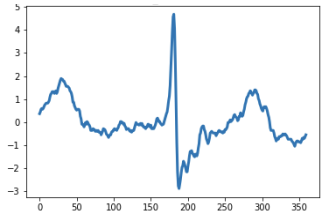
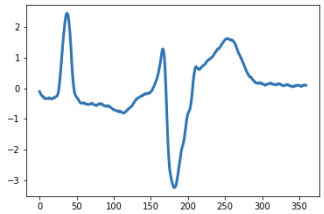
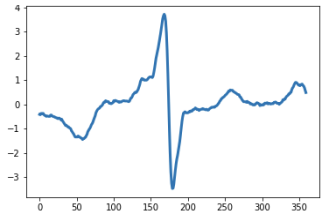
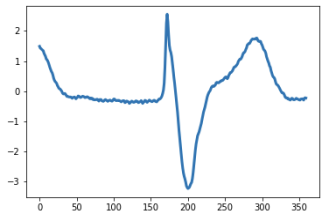
Category	AAMI Classes	Annotation	ECG Sample
N	Normal Beat	Normal Left/Right bundle branch block Atrial escape Nodal escape	
S	Supra-Ventricular Beat	Atrial premature Aberrant atrial premature Nodal premature Supra-ventricular premature	
V	Ventricular Ectopic Beat	Premature ventricular contraction Ventricular escape	
F	Fusion of Ventricular and Normal	Fusion of ventricular and normal	
Q	Unclassifiable Beat	Paced Fusion of paced and normal Unclassifiable	

Table 2.

Device	Raspberry Pi v2 	Huawei P30 Pro 
Chipset	Broadcom BCM2836	Kirin 980
CPU	ARM Cortex A53	ARM Cortex A55/A76
Mean Inference Time	4.76 ms	7.65 ms

Table 3.

Class	Original neural network				Quantized neural network			
	Acc	*Sen*	*Spe*	F_1	*Acc*	*Sen*	*Spe*	F_1
N	99.9%	99.8%	99.9%	99.8%	99.4%	99.8%	99.3%	98.6%
S	99.9%	99.9%	99.9%	99.8%	99.7%	99.1%	99.9%	99.4%
V	99.9%	100%	99.9%	99.9%	99.6%	99.5%	99.7%	99.3%
F	99.9%	99.7%	100%	99.8%	99.5%	99.4%	99.9%	96.7%
Q	99.9%	99.9%	100%	99.9%	99.9%	99.9%	100%	99.9%

Table 4.

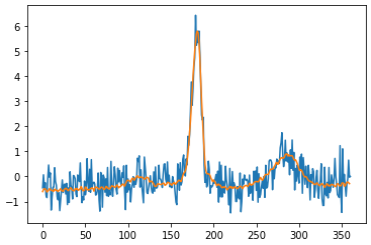
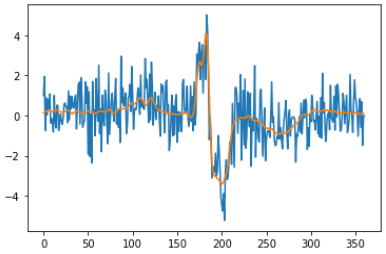
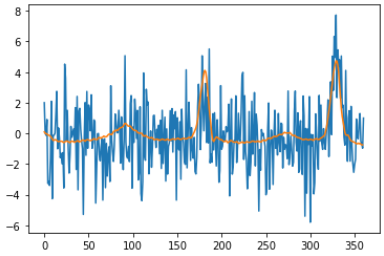
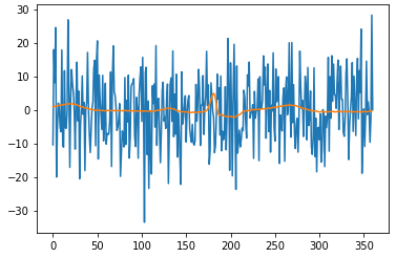
Noise / Artifact	Signal Sample
Electrode contact noise	
Power line interference	
Instrumentation noise	
Motion / Muscle contraction	

Table 5.

Work	*Acc*	*Sen*	*Spe*	*Ppr*	F_1	Type
Kachuee *et al.* [39]	93.4%	-	-	-	-	CNN
Acharya *et al.* [8]	93.5%	96.0%	91.6%	97.87	-	CNN
Martis *et al.* [40]	93.8%	99.5%	97.4%	99.1%	-	DWT + SVM
Li *et al.* [41]	94.6%	-	-	-	-	DWT + Random Forest
Martis *et al.* [42]	99.3%	99.9%	99.8%	99.2%	-	Probabilistic NN
Proposed	**99.6%**	**98.5%**	**99.8%**	**99.2%**	**98.8%**	**CNN**
Proposed / 20% noise	**96.4%**	**89.3%**	**97.6%**	**94.2%**	**90.6%**	**CNN**
Proposed / 50% noise	**91.9%**	**77.5%**	**94.8%**	**88.8%**	**80.2%**	**CNN**
Proposed / 100% noise	**86.7%**	**62.1%**	**91.4%**	**85.5%**	**65.1%**	**CNN**