Analysis of the Machine Learning Classification of Cardiac Disease on Embedded Systems

Sanda Thura
Computer Science
James Madison University
Harrisonburg, VA 22807
thurasx@dukes.jmu.edu

Jason Forsyth
Engineering
James Madison University
Harrisonburg, VA 22807
forsy2jb@jmu.edu

Kevin Molloy
Computer Science
James Madison University
Harrisonburg, VA 22807
molloykp@jmu.edu

Jacob Couch
Applied Physics Laboratory
Johns Hopkins University
Laurel, MD 20723
jacob.couch@jhuapl.edu

***Abstract*—Heart disease has become a major global health concern that is affecting millions of people worldwide. The situation is particularly critical in developing countries where the access to medical facilities is limited. This barrier to health care leads to increased fatalities from heart disease. Early diagnosis of cardiovascular conditions can be lifesaving. However, personal medical-grade equipment can be expensive and not easily accessible for people living in these areas. It is important to expand the same level of medical care to these communities at an affordable price. Our research aims to investigate the performance of a machine learning model on a low-cost embedded system. This study will evaluate the accuracy, run time, and overall performance of the model in diagnosing cardiovascular diseases. The results will help us determine the feasibility of using machine learning models for classifying cardiovascular disease in low-cost embedded systems. A selected machine learning model has been trained, modified, and compiled into the embedded system. The model returns the classification results based on preprocessed input data. Multiple metrics are collected to measure the performance of the model and the embedded system. The preliminary results are promising with accuracy levels similar to the original model. If these results hold up in multiple trials, it is expected that the machine learning model for classifying cardiovascular diseases on the embedded system will be practical and useful in extending affordable medical care to developing countries.**

***Index Terms*—neural networks, machine learning, diagnosis, electrocardiogram**

I. Introduction

The World health organization (WHO) estimates that cardiovascular diseases (CVDs) have been the leading fatal disease worldwide that contributes to 17.9 million fatalities annually [1]. CVDs may lead to potentially life-threatening complications if they are not diagnosed early or left untreated. In developing countries, people often have limited access to medical facilities, doctors, and primary care compared to developed countries [1]. The doctor-to-patient ratio in developing countries is significantly lower compared to developed countries. In addition to the limited access to medical care, the financial situation is also a barrier to a healthy life. Daily minimum income in developing countries is significantly low compared to developed countries. Furthermore, lifestyle and diet contribute to a higher chance of getting CVDs in Asian countries [2], [3].

An affordable and accurate wearable device for the classification of heart diseases can provide a promising alternative for individuals who have limited access to expensive healthcare technologies. With early detection of heart disease, individuals can seek timely medical intervention and prevent potentially catastrophic outcomes. Therefore, the development and implementation of affordable wearable devices for the accurate classification of heart diseases can have significant implications for public health. While there are multiple approaches to diagnosing heart disease, our approach will focus on utilizing powerful deep learning methods to accurately detect instances of CVDs. More specifically, we will utilize convolutional neural networks (CNN) to detect arterial fibrillation (AF), an indicator of coronary artery disease (CAD), a specific disease in the CVD family. To perform this detection we will leverage the ability to transform electrocardiography (ECG) measurements into "images" that can be easily classified by CNNs. While these CNN models are often resource intensive and developed on traditional desktop computers, to show that our work could be deployed in the developing world, we will explore how these CNN models perform after being implemented on a low-cost embedded system. Notably we will examine the impact on classification accuracy and run-time performance arterial fibrillation detection.

Overall, our research results indicate that the CNN model is able to classify CAD on the low-cost embedded system with near 80% accuracy. Although we witnessed increased classification time compared to traditional desktop processing, we have proven that the model can perform the complex computation within the constraints of an embedded system with little to no loss of accuracy. Throughout this paper, we will discuss future research areas, opportunities and expected challenges for future researchers.

The remainder of this document is organized as follows: Section II discusses related work cardiac health with a specific focus on the prevalence of CVD in developing countries and highlights the critical need for an effective and affordable solution to address cardiac disease in developing countries. Section III outlines our research methodology and decisions for each component of the system's development. Section IV discusses the analysis of the data collected and the evaluation of the developed model's performance. Lastly, Section V summarizes our work and provides suggestions for future

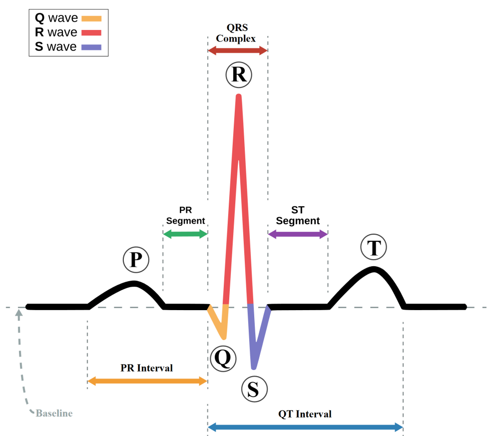

Fig. 1: ECG beat segment for Sinus Rhythm [6]

researchers.

II. Background and Motivation

In this section, we discuss the utility of ECGs, the prevalence of CVDs in developing countries and the limited access to medical facility they experience, and related work in automated classification of heart disease.

A. Overview of Cardiovascular Disease and Detection

Cardiovascular diseases are a group of heart disorders that affect the blood circulation in the body [1]. They are one of the leading causes of death globally. CVDs contains a wide variety of conditions, including but not limited to coronary heart disease, stroke, peripheral arterial disease, and aortic disease. These diseases can lead to serious, life-threatening complications if they are not diagnosed in an early manner or treated appropriately.

Generally, there exists a correlation between a person having coronary artery disease and the presence of atrial fibrillation(AF) [4]. AF is an irregular heart rhythm that can be easily detected on ECG waves. If a person has AF, there is a higher chance that the person has CAD [4]. A typical ECG wave is shown in Figure 1 where individual components of the heart beat are annotated as P, Q, R, S, T waves. From the ECG waves, we can find out an irregular and rapid heartbeat and an irregular rhythm within RR interval, the time between R peaks of two consecutive beats, but with no discernible P waves. The diagnosis of AF on ECG can be confirmed by observing this pattern on any ECG leads. In some cases, a longer monitoring period may be necessary to capture intermittent episodes of AF [5].

B. Prevalence of Cardiovascular Disease in the Developing World

Undiagnosed or untreated CVDs can result in severe and life-threatening complications such as heart attacks and even sudden death [7]. Access to medical facilities and healthcare professionals is a challenge for individuals in developing countries [8]. This is mainly due to the fact that the doctor-to-patient ratio is significantly lower than that of developed countries as shown in Table I. Data from the 2018 City Life Survey indicates that 37% people living in Yangon, the largest former capital city in Myanmar with a population of 5 million, do not have enough financial power to pay for an unexpected medical emergency [9]. 44% of people living in Yangon also find it hard to afford health care along with 33% in Mandalay and 48% in Mawlamyine, which are capital cities in different states.

TABLE I: Average Physician to patient ratio in the developed countries and developing countries [10] [11] [12]

Country	Physician-patient ratio (per 1000)	Year
United States	2.95	2016
United States	2.604	2018
France	6.53	2018
Thailand	0.92	2019
Vietnam	0.83	2016
Myanmar	0.7	2018
Myanmar	0.37	2019

In addition, the financial situation in developing countries presents a further barrier to obtaining medical care. The daily minimum income is much lower than in developed countries. A report from the Myanmar ministry of health stated the average pay for office staff as 150,000 Myanmar Kyats (MMK) (about 90 USD) per month [13]. An online informal survey was conducted to assess low-income to middle income people willingness to spend on the medical device for regular screening. It is found that the average they are willing to spend is about 80,000 MMK to 100,000 MMK (about 40 to 60 USD) [14]. Research conducted in 2017 indicated that the cost for a hospital visit can range approximately from 28,000 MMK to 1,961,806 MMK (27 USD to 1900 USD) [13].

Research shows that individuals in Myanmar are particularly susceptible to coronary artery disease and rheumatic disease [15]. While the latter requires medical professionals for diagnosis, coronary artery disease can be detected via electrocardiogram (ECG) and diagnosed with machine learning algorithms on wearable devices. Thus, the development of a low-cost smart device that enables early diagnosis of coronary artery disease (CAD) has the potential to save millions of lives in developing countries. Early detection and treatment can be life-saving for individuals in these regions.

C. Related Work in Automated Detection of CVD

In recent years, there have been numerous attempts in utilizing machine learning algorithms for the diagnosis of CVDs on wearable devices [16]. Many machine learning algorithms have been developed and tested, with some demonstrating high levels of accuracy in classifying CVDs. For example, some algorithms have shown accuracy rates of up to 92% in detecting CVDs [17].

While our approach utilizes convolutional neural networks, there are other machine learning techniques that are used in CVD classification. One approach employs a support vector machine (SVM) to clasify for atherosclerosis and early CAD

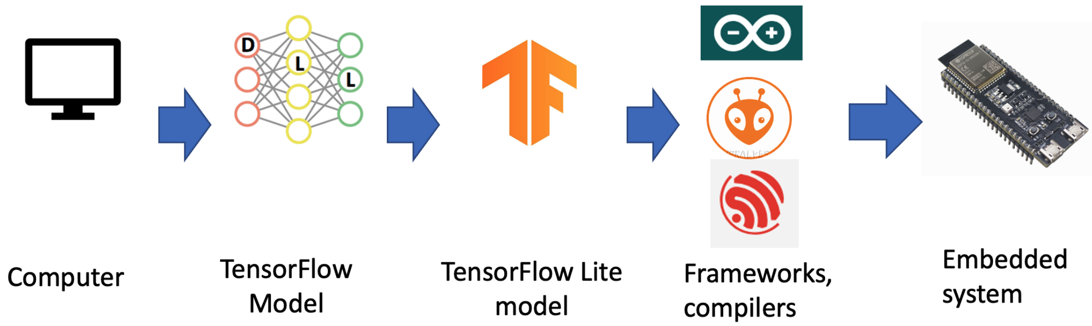

Fig. 2: Model Implementation Pipeline Showing the Progression from Desktop to Embedded System Implementation

detection using ECG signal from a single lead [18]. This algorithm has an overall accuracy of 88% after extracting the features and filtering the signals. Another approach is to use machine learning methods to analyze the features extracted from single lead ECG wave which are RR and QT interval time-series and ST-T segment waveforms [19]. RR interval is the time between R peaks of two consecutive beats. QT segments is the interval between the start of Q wave and the end of T wave. ST-T interval is the time between start of S wave to the end of T wave as seen in Figure 1 This approach has achieved an accuracy of 96% with 2 combined classifiers.

III. Methodology and Experimental Design

In this section we outline the major design decisions in our system, the development process and tools used, and the data collected in our experiments.

A. Major Design Decisions

When implementing our CVD detection pipeline there are several interrelated decisions to make. First is to find an example ECG dataset recorded with labels so a machine learning model can be trained. Second, the selected model must perform suitably well on that data and be reliable in a software framework to test its performance. Finally, as our goal is to understand the embedded system's performance, a target platform must be selected that can implement the machine learning algorithm. We discuss these factors in the remainder of the section.

1) Datasets: PubMed research in 2017 recorded 3.046 million new instances of atrial fibrillation globally, increasing 33% in the last 20 years [2]. Currently the worldwide prevalence of atrial fibrillation is estimated to be around 37.6 million cases, which is only 0.51% of the worldwide population. We selected a dataset that includes a variety of ECG data, including about 8,000 recordings of normal waves, atrial fibrillation, unclassified recordings, and other cardiac waves from the short single lead ECG Recording of the 2017 PhysioNet Cardiology Challenge [20]. This ECG dataset contains 5076 normal waves, 2415 atrial fibrillation (AF), 279 unclassified and 758 others recordings. Unclassified recording means that the ECG waves cannot be classified into any known ECG categories. These are mostly noise and a doctor would order a new ECG recording. Other means the ECG could be any cardiac waves other than normal, atrial fibrillation or noise.

2) Classification Algorithm: We will select an algorithm that can classify atrial fibrillation from single lead ECG records. We will reject algorithms that are developed for 12 lead ECG data since our focus of the research is to develop a system that utilize single lead ECG module. We chose single lead ECG since the state of the art device in commercial wearable device only has single lead ECG ability such as Apple Watch. At this time of research, we will not consider algorithms that uses previous stored data such as k-nearest neighbors (KNN) due to computational resource limitations on the embedded system. Thus, we selected an machine learning algorithm from this paper "Convolutional Recurrent Neural Networks for Electrocardiogram Classification" [21]. The algorithm contains a preprocessing step to filter, denoise, and standardize the data, then convert it into spectrogram images for classification. The raw ECG data is from Physionet Computing in Cardiology Challenge 2017 [20].

3) Machine Learning Frameworks: When selecting a machine learning framework, we considered multiple factors, such as support for desktop and embedded implementation, compatibility with our embedded system, and support for our machine learning model. We considered several frameworks, including TensorFlow and TensorFlow Lite, PyTorch, PyTorch Mobile, and ARM CMSIS-NN. We selected TensorFlow and TensorFlow Lite for Microcontrollers since they offer support for complex machine learning functions and officially support the Arduino and ESP32 frameworks [22]. TensorFlow also provides end-to-end support for both desktop-level and microcontroller-level machine learning models, reducing the implementation time and developmental errors. Although TensorFlow Lite for Microcontroller has some limitations, it supports most of the operations needed for our research.

4) Embedded Systems: During the implementation of our machine learning model, on the first development embedded system, we discovered that the model size was approximately 2 MegaBytes (MB); thus it would require at least 3MB of RAM for computation and to store the model and ECG sample data. Therefore, for our research purposes, we decided to use a low-cost embedded system with sufficient RAM necessary. We selected the "ESP32 S3 Devkit C1" board as our embedded system due to several reasons as TensorFlow offers official support for implementing machine learning models on the ESP32 and ESP32 boards offer sufficient memory, with up to 8MB of RAM, allowing for us to test a wide range of machine learning models. Furthermore, the ESP32 supports 32-bit floating-point calculations, which are essential for machine learning inferences. For embedded systems developmental frameworks, this work used the Platform I/O and ESP-IDF frameworks [23], [24].

B. Development Process and Tools

Collecting together the design choices in Section III-A, Figure 2 shows the set of software frameworks and hardware platforms used to implement our system. The section will discuss the particulars of that pipeline and how data flows from one element to another.

Construction of the machine learning model begins on the desktop. The raw ECG data from Physionet [20] is converted into a spectrogram as a pre-processing step based upon the model in the paper [21]. This CNN model is then trained on the dataset (with 70-20-10 split where 70% of data is used for training, 20% is used for validation and 10% for testing) while hyperparameters (number of convolutional layers, functions) are tuned to achieve the best model performance in terms of accuracy and F1 score. The desktop development work is performed using the TensorFlow framework using the Python language.

Once the model is trained, we save it as a full-sized TensorFlow model and convert it into a TensorFlow Lite model. Then, we serialize the TensorFlow Lite model into a Flatbuffer C++ byte array or a TensorFlow Lite model for microcontrollers [25], which is loaded into the embedded system using the Platform I/O and ESP-IDF frameworks. On the embedded system, the model is reassembled from the Flatbuffer byte array. After the model is loaded, the ECG spectrogram can be sent to the embedded system over a serial port to perform classification.

C. Experimental Setup

In our research, we are interested to learn the accuracy trend, inference time, and model complexity of our machine learning model on the embedded system and on the desktop computer. To achieve this, we take a two-pronged approach. First, to save time on preprocessing data in the embedded system, we preprocess the testing data on the computer, which can process multiple ECG testing records in less than a second. The testing data is a separate dataset that has not been seen by the model during training. Each testing data spectrogram is then passed down to the embedded system through the serial communication channel between the host computer and the board. The embedded system then classifies each spectrogram and reports the results back to the host computer, where they are aggregated. The data for each record includes the classification result on the embedded system and the classification time and the data transport time. To validate the pipeline that includes the embedded system, another experiment is performed where all the processing is performed on a single desktop computer. We also run the full-sized model and the TensorFlow lite model on the computer to classify the same testing dataset, and their metrics are collected. We test multiple variants of the same models to see the trend in performance metrics such as accuracy, classification time and model size.

IV. Discussion and Results

A. Results

As described in Section III-B, multiple iterations on the CNN models were created to tune and identify the best result on our target. Table II shows the modifications made to different models during training to achieve similar accuracy to the source work [21]. Each model is named sequentially on model version with changes to the number of layers, blocks, and other modifications to the CNN architecture. The other columns indicate the accuracy from each model on the "Desktop" results from the TensorFlow framework. Overall, the best model was determined to be CNN_ECG_10 with an accuracy of 81.5% and F1 score of 81.5%. The confusion matrix for this model shown in Figure 3 indicates that the current trained model can classify "Other" and "Noise" very well. While the model shows moderate classification performance for "Normal" and "Atrial fibrillation", it is occasionally prone to misclassification between these two classes. We see this trend throughout the different model training. We conclude that some data may be very similar to differentiate between two classes. After identifying the best desktop model, we implemented that machine learning model to the embedded system following the process in Section III-B.

The embedded system implementation was equivalent in accuracy when classifying the dataset, with an accuracy and F1 of 80.8% and 80.9%, respectively. However, run-time performance was significantly impacted. While the desktop model can perform many classifications per second, the embedded system could only perform a classification every ∼2 minutes. While a slower classification time was expected, this much lower speed will have an impact on the overall suitability of the embedded system for real-world use.

TABLE II: Model Modifications Table

Name	Layers	Block	Modifications	Accuracy	F1
cnn_ecg_1	depth 4	block 1		25.0	10.0
cnn_ecg_2	depth 2	block 4	no mask layer	71.9	72.3
cnn_ecg_3	depth 4	block 2		73.2	73.5
cnn_ecg_4	depth 2	block 2		48.6	48.6
cnn_ecg_5	depth 1	block 4		59.2	58.9
cnn_ecg_6	depth 1	block 3		45.1	45.1
cnn_ecg_7	depth 3	block 1		43.7	40.2
cnn_ecg_8	depth 2	block 3	no mask, no means layer	25.0	10.0
cnn_ecg_9	depth 2	block 4	no mask layer	76.3	76.2
cnn_ecg_10	depth 2	block 4	no mask layer, balanced data	81.5	81.5

B. Discussion

In this section we discuss the implementations of our results and determine whether our implementation has sufficient

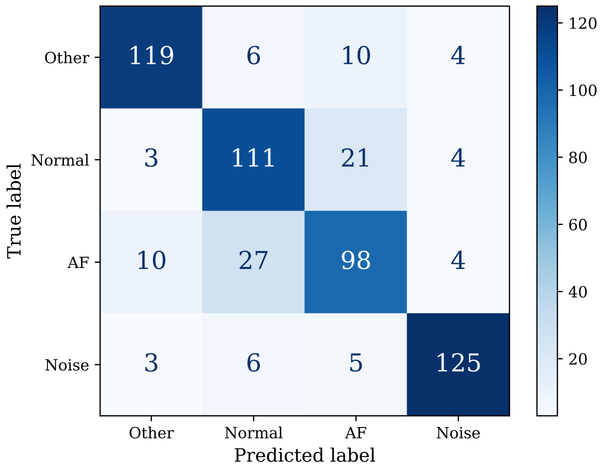

Fig. 3: Confusion Matrix of Best Performing Model: CNN_ECG_10

accuracy and run-time performance to operate effectively as a diagnostic tool in developing countries.

1) Accuracy: There is no single standard or guideline for the minimum required accuracy of a machine learning model for classifying heart diseases, as it may depend on various factors such as the severity and complexity of the diseases being classified, the intended use of the model, and the acceptable level of false positives and false negatives. However, studies in the Bayoumy paper have shown that many research models achieve an accuracy of around 80-90% [16]. Our approach barely achieves this threshold with an 80% classification accuracy on the embedded system as shown in TableII.

While our overall accuracy is near this threshold, that metric is across all classes in the dataset, and given our mission to be a diagnostic tool, our performance detecting and rejecting AF should be explored in more detail. Specifically, we will examine our sensitivity, which indicates our true positive rate which is our success at correctly identifying AF, and our specificity, which indicates our true negative rate of successfully rejecting individuals not experiencing AF. These numbers are important because accurate detection of AF avoids unnecessary medical costs for users if they are alerted of AF when they do not have the condition. Conversely, it is important to alert users who have the condition for prompt medical attention.

We currently achieved a true positive rate (sensitivity) of 70.5% for the AF class and a true negative rate (specificity) of 91.4% to AF class. While we have high true positive rate, we are still on the lower end for the true positive rate. Ideally, we should achieve sensitivity of 100%. With current low true positive, if a user has an actual AF episode, the model may miss it or wrongly classify the event. Even with prolong continuous use of the device and assuming we captured the AF event during the ECG recording, we only have 70% confidence that the result is correct. There may also a condition where we may not captured the AF episode during 60 seconds ECG recording window due to the uncertainty of AF events and AF stages in different person [26].

In attempting to improve our accuracy, we tried several approaches. Firstly, we developed our own custom loss function that penalizes wrong class classification more heavily. We hypothesized that this would lead to more accurate classification results. Additionally, we reduced the problem from multi-class classification to binary classification (i.e., predicting whether or not heart disease is present). Next, we experimented with adding or reducing the model's layers to understand if the classification results may improve within the embedded system's resource limitations. However, despite our best efforts, we did not observe any significant improvements in the model's accuracy. Based on our analysis of these experiments, we believe that the limited size and scarcity of our training data set may have been the bottleneck that prevented us from achieving greater accuracy in the model. It is possible that a larger and more diverse training data set would have enabled us to discover better modifications to the model that would have improved its accuracy.

2) Run-time Performance: We noticed the increase inference time when the model classify the data. It is not ideal for daily use. Currently, the model takes 0.8 seconds to classify 556 recordings on the desktop computer, however the same model will take about 61200 seconds on the embedded system. This implies that a single 60 seconds ECG record takes nearly 2 minutes to process. If the AF events happens during the processing time or classification time, a user may not know the actual result. Thus, we can only sample one-third of the time the device is operational. Combined with our lower accuracy, this poor sampling rate negatively affects the chances of correctly capturing, and subsequently identifying, an AF event. As we are missing data two-third of the time, our true positive will be reduced as the AF event may not be captured in the recording.

Given the very long inference times experienced, we conducted a series of experiments to determine what elements of the microprocessor caused this drop in performance.. We considered two main areas: the impact of floating point calculations and the placement of data in internal or external memory. No impact was noted on the difference in float point instructions. However, there was a significant impact on whether the TensorFlow Lite model memory was placed in the internal or external memory of the ESP32 system. The bottleneck between the internal CPU and external RAM caused the model to experience a long waiting time when accessing data in the tensor memory. We delved deeper into the bottleneck and found out that the slower clock speed on the memory may be a factor. The CPU core's clock speed is approximately 240 MHz, while the external RAM's clock speed is only 40 MHz. Consequently, the CPU may have already completed its calculation, but it could be waiting for new data to arrive through the data bus. To mitigate the longer inference times, we investigate various optimization techniques for improving the architecture of the development used in our research. These techniques include optimizing the data pipeline and employing memory allocation strategies that balance the data's proximity to the CPU core and external RAM, ensuring that the data is easily accessible to the model without causing any delays. We will need more robust data to make a conclusion if this will improve the overall performance of the system.

V. Conclusions and Future Work

Our research examined the performance of a machine learning model for detecting CVDs on an embedded system. We discovered that the accuracy of the model remained consistent across different platforms. However, we observed a decrease in the runtime performance of the model on the embedded system. We may see the changes in the runtime performance if we implement the preprocessing algorithm on the board. We attribute this long inference time to the limitations of the embedded system, which we expect to improve as low-cost microcontroller technology advances. Additionally, we noticed the limitations of the TensorFlow framework on embedded systems, but we anticipate that it will support more operations in the future. We used a very limited dataset to train the model which is also used by the selected model developer. This limited dataset may also be a bottleneck for increasing accuracy of the model.

Going forward, future researchers should consider the following suggestions. For machine learning researchers, there are more direct approaches to analyze CAD such as analyzing the ST-T segment changes for each heart beat using CNN, recurrent neural network (RNN) or other applicable model. The simpler the preprocessed data to classify, the more accurate and faster it will be on the embedded system. It is also suggested to use larger and more diverse dataset to train the model such as PTB-XL dataset [27]. For engineers, it is suggested to implement the hardware level ECG data recorder and preprocessor to improve the total time for each instances. For production of low-cost devices, it is also needed to record the total power consumption for daily uses and optimize accordingly. Overall, this work has shown it is feasible to implement a machine learning model on the embedded system at little to no cost of accuracy loss. When these machine learning driven low-cost devices are into mass-productions, it has a potential to save countless lives to people all over the world.

References

[1] World Health Organization, "Cardiovascular diseases (cvds)," https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds), 2021, accessed: February 8, 2023.

[2] G. Lippi, F. Sanchis-Gomar, and G. Cervellin, "Global epidemiology of atrial fibrillation: an increasing epidemic and public health challenge," *International Journal of Stroke*, vol. 16, no. 2, pp. 217–221, 2021.

[3] H. Ueshima, A. Sekikawa, K. Miura, T. C. Turin, N. Takashima, Y. Kita, M. Watanabe, A. Kadota, N. Okuda, T. Kadowaki, and et al., "Cardiovascular disease and risk factors in asia," *Circulation*, vol. 118, no. 25, p. 2702–2709, 2008.

[4] F. Liang and Y. Wang, "Coronary heart disease and atrial fibrillation: a vicious cycle," *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 320, no. 1, pp. H1–H12, 2021, pMID: 33185113. [Online]. Available: https://doi.org/10.1152/ajpheart.00702.2020

[5] NHS. (accessed 2023) Atrial fibrillation. NHS. [Online]. Available: https://www.nhs.uk/conditions/atrial-fibrillation/

[6] A. A. Atkielski), "Schematic diagram of normal sinus rhythm for a human heart as seen on ecg," https://commons.wikimedia.org/wiki/File:SinusRhythmLabels.svg, January 13 2007.

[7] C. Clinic, "Cardiovascular disease," Accessed on February 8, 2023. [Online]. Available: https://my.clevelandclinic.org/health/diseases/21493-cardiovascular-disease

[8] C. Article, "Improving access to quality health care in developing countries," *Philanthropy News Digest*, January 2019. [Online]. Available: https://givingcompass.org/article/improving-access-to-quality-health-care-in-developing-countries

[9] The Asia Foundation, *Insights into urban well-being in Myanmar: The 2018 City Life Survey*, 2018.

[10] "Physicians (per 1,000 people)," 2018. [Online]. Available: https://data.worldbank.org/indicator/SH.MED.PHYS.ZS

[11] A. Young, H. J. Chaudhry, X. Pei, K. Arnhart, M. Dugan, and G. B. Snyder, *A Census of Actively Licensed Physicians in the United States, 2016*, vol. 103, 2017.

[12] Y. M. Saw, T. M. Than, Y. Thaung, S. Aung, L. Wen-Shuan Shiao, E. M. Win, M. Khaing, N. A. Tun, S. Iriyama, H. H. Win, and et al., "Myanmar's human resources for health: Current situation and its challenges," *Heliyon*, vol. 5, no. 3, 2019.

[13] , *Myanmar Ministry of Health 3 Years Improvements*. Myanmar Ministry of Health, May 2019.

[14] S. Thura, "Personal communications with people on 3rd march 2022," 2022.

[15] E. E. Swe, K. K. K. Htet, P. Thekkur, L. L. Aung, L. L. Aye, and T. Myint, "Increasing trends in admissions due to non-communicable diseases over 2012 to 2017: findings from three large cities in myanmar," *Tropical Medicine and Health*, vol. 48, no. 1, pp. 1–12, 2020.

[16] K. Bayoumy, M. Gaber, A. Elshafeey, O. Mhaimeed, E. H. Dineen, F. A. Marvel, S. S. Martin, E. D. Muse, M. P. Turakhia, K. G. Tarakji *et al.*, "Smart wearable devices in cardiovascular care: where we are and how to move forward," *Nature Reviews Cardiology*, vol. 18, no. 8, pp. 581–599, 2021.

[17] E. Tiryaki, A. Sonawane, and L. Tamil, "Real-time cnn based st depression episode detection using single-lead ecg," in *2021 22nd International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2021, pp. 566–570.

[18] A. Kaveh and W. Chung, "Automated classification of coronary atherosclerosis using single lead ecg," in *2013 IEEE Conference on Wireless Sensor (ICWISE)*, 2013, pp. 108–113.

[19] L. Yao, C. Liu, P. Li, J. Wang, Y. Liu, W. Li, X. Wang, H. Li, and H. Zhang, "Enhanced automated diagnosis of coronary artery disease using features extracted from qt interval time series and st–t waveform," *IEEE Access*, vol. 8, pp. 129 510–129 524, 2020.

[20] PhysioNet, "Physionet/cinc challenge 2017," https://physionet.org/challenge/2017/, 2017.

[21] M. Zihlmann, D. Perekrestenko, and M. Tschannen, "Convolutional recurrent neural networks for electrocardiogram classification," in *2017 Computing in Cardiology (CinC)*. IEEE, 2017, pp. 1–4.

[22] TensorFlow, "Tensorflow lite for microcontrollers," https://www.tensorflow.org/lite/microcontrollers, 2023.

[23] PlatformIO, "Platformio," 2023. [Online]. Available: https://platformio.org

[24] Espressif, "ESP-IDF — Espressif IoT development framework," https://www.espressif.com/en/products/sdks/esp-idf, 2023, [Online; accessed 8-Feb-2023].

[25] Google, "Flatbuffers," 2023. [Online]. Available: https://github.com/google/flatbuffers

[26] "Types of atrial fibrillation," https://www.nhlbi.nih.gov/health/atrial-fibrillation/types, 2023, accessed: February 8, 2023.

[27] P. Wagner, N. Strodthoff, R.-D. Bousseljot, D. Kreiseler, F. I. Lunze, W. Samek, and T. Schaeffter, "Ptb-xl, a large publicly available electrocardiography dataset," *Scientific data*, vol. 7, no. 1, p. 154, 2020.