

A Smart Health Application for Real-time Cardiac Disease Detection and Diagnosis Using Machine Learning on ECG Data

Ucchwas Talukder Utsha¹, I Hua Tsai², and Bashir I. Morshed³

¹ uutsha@ttu.edu

² i-hua.tsai@ttu.edu

³ bmorshed@ttu.edu

Department of Computer Science, Texas Tech University

Abstract. Cardiac disease, also referred to as cardiovascular disease, is a collection of conditions that affect the heart and blood vessels. Medical professionals typically use a combination of medical history, physical examination, and various diagnostic tests, such as electrocardiograms (ECG/EKG), echocardiograms, and stress tests, to diagnose cardiac diseases. In response to this issue, we are introducing a mobile application that continuously monitors electrocardiogram signals and displays both average and instantaneous heart rates. The aim of this project is to detect and diagnose cardiac diseases so that patients can become informed about their heart health and take appropriate actions based on the results obtained. To identify diseases from real-time ECG data, we used machine learning (ML) classifiers and compared them with offline data to validate the classification. The model we used in our application is pre-trained on the MIT-BIH Arrhythmia Database, which contains a wide range of heart conditions. We used Artificial Neural Network (ANN) as a pre-trained model for multiclass detection as it performed the best among ML models, showing an overall accuracy of 94%. The performance of the model is evaluated by testing it on the application using MIT-BIH test Dataset. On the application, the beat-detecting pre-trained model showed an overall accuracy of 91.178%. The results indicate that the Smart-Health application can accurately classify heart diseases, providing an effective tool for early detection and monitoring of cardiac conditions.

Keywords: Cardiac disease · Electrocardiograms · Pre-Trained Model · Smart-Health Application.

1 Introduction

Cardiovascular diseases (CVDs) are critical and common heart diseases that can be detected using electrocardiogram (ECG or EKG) signals. The ECG signals are used to diagnose different types of heart diseases such as heart failure, myocardial infarction (MI), premature ventricular contractions (PVCs), etc. Analyzing the bio-electrical signals of each heartbeat, cardiologists can detect abnormalities in the heart, such as irregular heartbeats or abnormal rhythms. However, manual scrutiny of continuous ECG signals for long durations for each patient is not practical or feasible [1]. Thus, automated detection using machine learning (ML) models is essential for accurate and efficient diagnosis of heart disease.

With the integration of IoT technology into heart disease monitoring, wearable devices and sHealth applications are gaining popularity [2]. These devices and apps use sensors and algorithms

to collect and analyze ECG signals in real time, providing patients with immediate feedback and promoting better management of heart disease. In addition to ML algorithms, the integration of IoT technology and sHealth applications has revolutionized the way we approach cardiovascular health. KardiaMobile is a compact, portable electrocardiogram (ECG) gadget that allows people to monitor their heart health and detect potential cardiac problems [3]. It works by recording a single-channel ECG through two electrodes on the device's back. It has been shown to be successful in clinical investigations for identifying atrial fibrillation (AF), with a sensitivity of 96.6% and a specificity of 94.1% [4]. An ECG check, like KardiaMobile, is a portable electrocardiogram (ECG) equipment that employs two or more electrodes to record the electrical activity of the heart [5]. The ECG check app transfers the recorded ECG data to a server for processing. Apple Watch is another device that can monitor the ECG signal and measure heart rate. The ECG feature, which is available on Apple Watch Series 4 and later, enables users to record an electrocardiogram, a test that examines the electrical activity of the heart. The watch can detect aberrant cardiac rhythms like AF and alert the user if one is identified [6]. Previously, our research team developed another smart health framework using body-worn flexible Inkjet-printed (IJP) sensors, commercial wearables such as smart wristbands, a scanner on a printed circuit board, and customized smartphone software [7]. The technology to collect and analyze ECG signals, providing patients with real-time feedback and enabling them to take control of their heart health using wearable devices and smart health apps has the potential to greatly improve the prevention, management, and treatment of heart diseases.

The use of multi-stage classification has shown significant potential in tackling the complexities of adjusting Artificial Intelligence (AI) models to novel sensor data or in the evolution of decision-making methodologies in smart systems [8]. Segmenting the AI model into various stages enhances its scalability and upgradability, offering a more flexible alternative compared to single-stage classification that could potentially struggle in adapting to changes. Multi-stage classification allows for independent modifications, enabling a more flexible and adaptable approach. This approach has shown great potential in the detection of cardiac diseases, demonstrating its scalability and upgradability for smart health systems. With the integration of AI into cardiac disease detection, multi-stage classification has emerged as a valuable tool in improving patient outcomes and promoting heart health. Its benefits include increased accuracy, efficiency, and the ability to adapt to changing data and circumstances. As such, multi-stage classification has the potential to revolutionize the field of smart health and transform the way we approach cardiac disease detection and management [9-15].

A multistage algorithm for automatic ECG data classification combines different procedures for dimensionality reduction, consensus clustering, and fast supervised classification algorithms [9]. Two multilayer perceptron (MLP) and one self-organizing map (SOM) networks perform better than using raw data or individual features for classifying six common ECG waveforms with an average recognition rate of 0.883 within a short training and testing time [10]. A multi-stage deep learning classification model for automatic arrhythmia classification using ECG waveforms and Second Order Difference Plot (SODP) features in discriminating five types of heartbeats from the MIT-BIH Arrhythmia Database [11-12].

A multistage pruning technique to reduce the computational complexity of Convolutional neural network (CNN) models used in ECG classification for real-time detection of arrhythmias. [13]. The method presented entails a multi-tiered process to ensure precise classification of arrhythmia, leveraging 12-lead surface ECGs. This comprehensive procedure includes three distinct phases of noise reduction, an innovative feature extraction approach, and a finely optimized classification

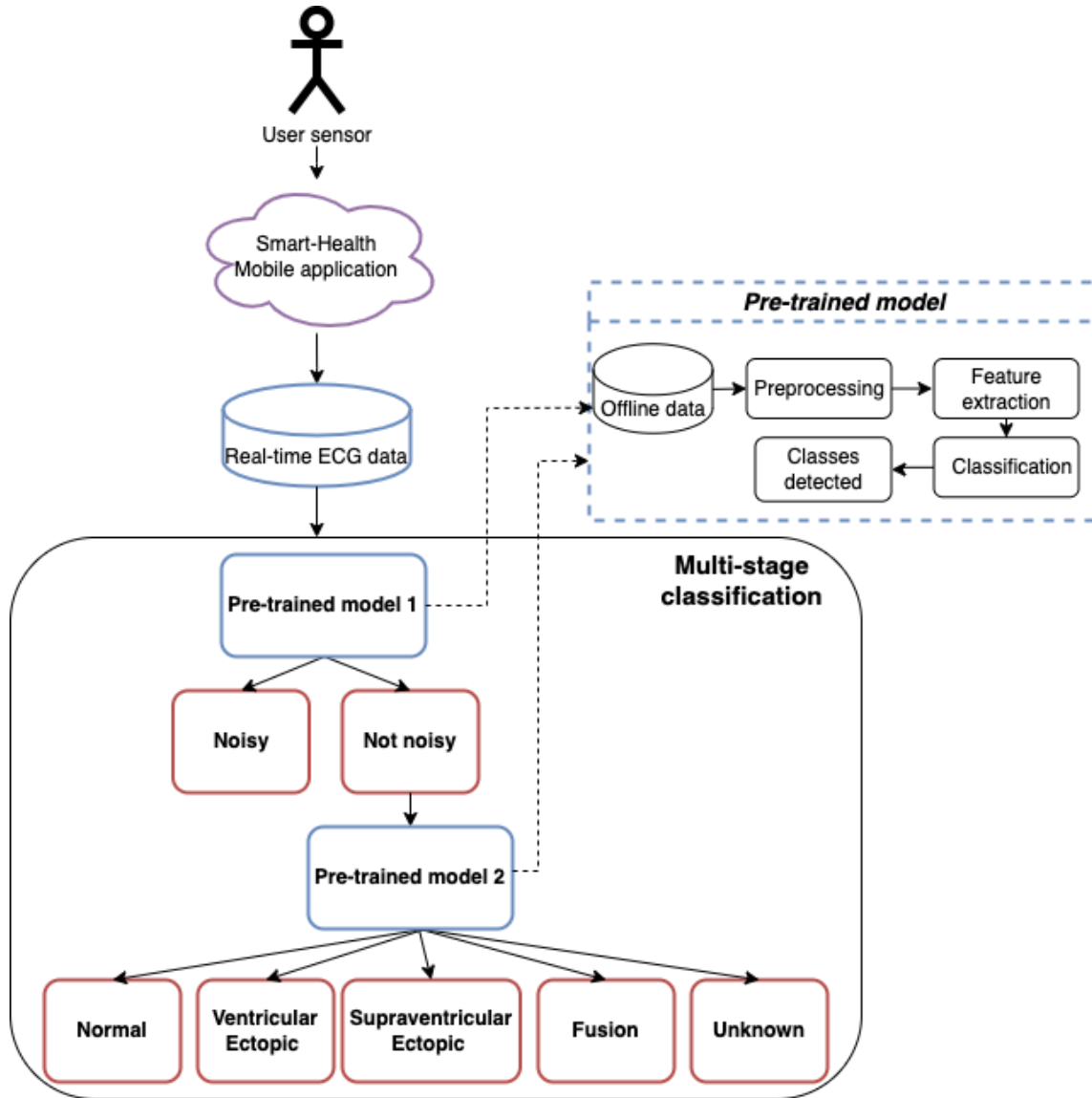


Fig. 1. The flowchart of the ECG signal processing, analysis, and classification.

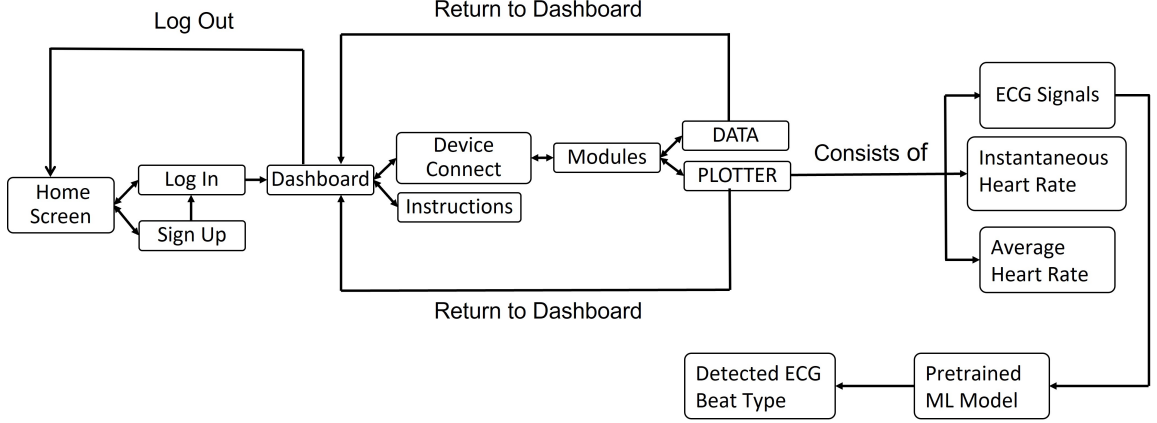


Fig. 2. Block Diagram of Smart-Health Application.

model.[14]. This technique employs features like dynamic amplitude range and autocorrelation maximum peak to identify and categorize various types of noise. [15].

This work presents a wearable ECG monitoring system that is cost-effective and capable of real-time monitoring through a smartphone application. The application provides real-time visualization of the ECG trace and heart rate detection, allowing for monitoring, assessment, and diagnosis. Also, we used Artificial Neural Network (ANN) as a pre-trained model for the application of disease detection. The main aspects of the proposed application include:

- It allows users to connect to an embedded system that collects ECG signals from wrists by electrodes via Bluetooth Low Energy (BLE) and monitor ECG signals on the screen of the application.
- The users can track their heart rate and rhythm over time, and share ECG recordings with healthcare providers for remote monitoring and diagnosis.
- The application can detect the Normal and Noisy signals from the ECG data.
- The application allows users to detect diseases from the incoming ECG signals.

2 METHODOLOGY

In this study, we utilized the MIT-BIH database, which follows the AAMI criteria to classify heart-beat types [16]. The MIT-BIH database consists of five categories of heartbeats, each containing multiple types of beats. Class N includes normal heartbeats, class SV includes Supraventricular Ectopic heartbeats, class V includes Ventricular Ectopic beats, class F includes Fusion beats and class Q includes Unknown beats. To achieve our goal, we accessed the PhysioNet database, an open-source public data resource, and selected the MIT-BIH arrhythmia database (mitdb) [17]. We divided the records into training and testing datasets. The offline data is then used in a Machine Learning model a pre-trained model for training before using real-time data on the Smart-Health application. The ECG signal processing involves a machine learning algorithm for preprocessing, analysis, and classification. The flowchart of the ECG signal processing is shown in Figure 1. Then,

we passed the MIT-BIH test dataset to the application to validate the pre-trained model on the application.

2.1 ECG data

ECG (electrocardiogram) data is a sort of medical data that captures the electrical activity of the heart. It is obtained by applying electrodes to the skin of the chest, arms, and legs and connecting them to a device that records and amplifies the electrical signals produced by the heart.

Signal processing and machine learning techniques are frequently used to analyze ECG data in order to extract diagnostic data and enhance clinical decision-making. Arrhythmia, myocardial infarction, and heart failure are just a few of the disorders that are frequently diagnosed and monitored using ECG data. Additionally, properties including heart rate variability, QT intervals, and P-wave morphology can be extracted from it. We collected ECG data using an AD8232 chip (Analog Devices, Wilmington, MA) implemented on our custom ECG data collection device [18]. Electrodes are attached to the wrists of the users and the other part of the electrodes are connected to Sparkfun nRF 52840 mini that is paired with the application via Bluetooth Low Energy (BLE) V5.3.

2.2 Application

The Smart-Health application is a mobile application that allows users to manage their health data. Figure 2 shows the block diagram of the Smart-Health Application.

Signal preprocessing: Signal preprocessing is an important step in analyzing ECG data and removing any noise or artifacts that may interfere with the accurate signal analysis. In the Smart-Health application, the ECG signals collected from the embedded system are preprocessed to remove noise. The frequency range of interest for ECG signals is between 0.5 Hz and 150 Hz [19]. The lower cutoff frequency of 0.5 Hz is chosen to remove any DC offset or drift in the signal, while the upper cutoff frequency of 150 Hz is chosen to remove any high-frequency noise or artifacts in the signal. The ECG signal is filtered on the application using a bandpass filter to remove any noise outside the frequency range of interest.

Peak detection and Heart Rate calculation: We used the Pan-Tompkins algorithm for peak detection of ECG signals which is a widely used method. First, we differentiated the filtered signal to emphasize the QRS complex's high-frequency components. To accentuate the QRS complex and reduce the T and P waves, we squared the differentiated signal. Then, we passed the squared signal to a moving window to produce a smooth envelope where a threshold is applied to detect the R-peaks. After that, we determined the heart rates from R-Peaks. In addition, we displayed the Average Heart Rate on the application to provide a more complete picture of the user's heart status. For that, we used the *Sliding Window* approach. We studied 30 heart rate measurements at the same time using a window size of 30. The same statistic was then computed for the next 30 data after adjusting the window by one heart rate value. As a result, users may simply monitor their heart rate and detect any abrupt changes.

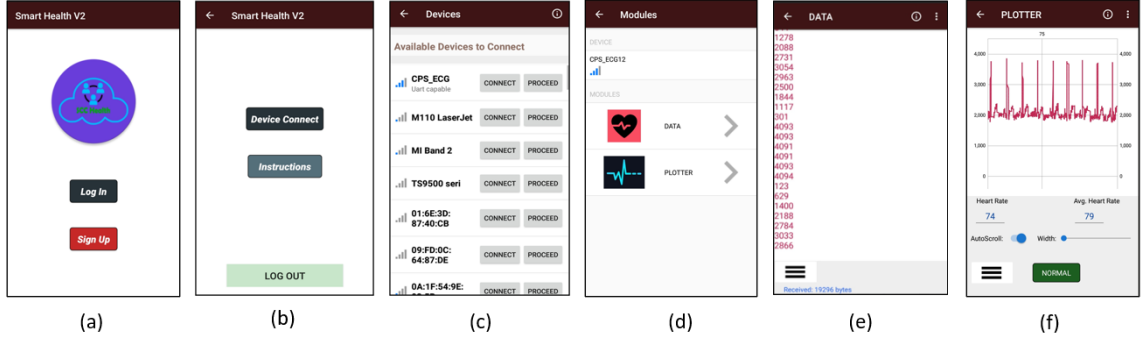


Fig. 3. Snapshots from the application (a) Home (b) Dashboard (c) Available Devices (d) Modules (e) Incoming Data (f) Real-time plot of ECG trace, Heart Rate and Signal type.

Pre-Trained Model on the application: We used pre-trained machine learning models in the Smart-Health application to diagnose medical conditions in real-time. We trained the models on the MIT-BIH dataset and then integrated them into the application.

To use a pre-trained machine learning model in our Android Studio Java application, we followed these steps :

1. **Train and Save the model:** First, we trained the model on a suitable dataset and saved it in a format that can be loaded by TensorFlow Lite. Here, we saved the model as a *.h5* file using the Keras model.save() method.
2. **Convert the model to TensorFlow Lite format:** Next, we converted the machine learning model to TensorFlow Lite format using the TensorFlow Lite converter. This produced a *.tflite* file that we used in our Android application.
3. **Add the model in the android application:** We loaded the *.tflite* file in the *assets* folder of the Android Studio project. Alternatively, we could go to the *File* → *Other* → *Tensorflow Lite Model* and import the *.tflite* file. It will be added in the *ml* folder on the project.
4. **Add dependency:** After that, we added the TensorFlow Lite interpreter dependency to the project's build.gradle file: *implementation 'org.tensorflow:tensorflow-lite:2.6.0'*.
5. **Load the model:** Then, we loaded the model from the *assets/ml* folder and created a *ByteBuffer* object to hold the input data.
6. **Get the predicted class:** Finally, we passed the ECG signal data through the interpreter to get the predicted class.

We validated the pre-trained model on the application by passing the offline dataset to it. That means, instead of real-time data, we passed the offline MIT-BIH test dataset to the application to find the exact label. Also, it acted like real-time data on the application. With this validation, we can now go for a clinical trial.

Real-time ECG check: Figure 3 shows some snapshots of the Smart-Health application. Initially, the user needs to register his or her details on the application. Then, s/he should *Log In* on the application. The various devices that can be connected through BLE connection are shown on the smartphone through the *Device Connect* button on the Dashboard screen. Then, the application

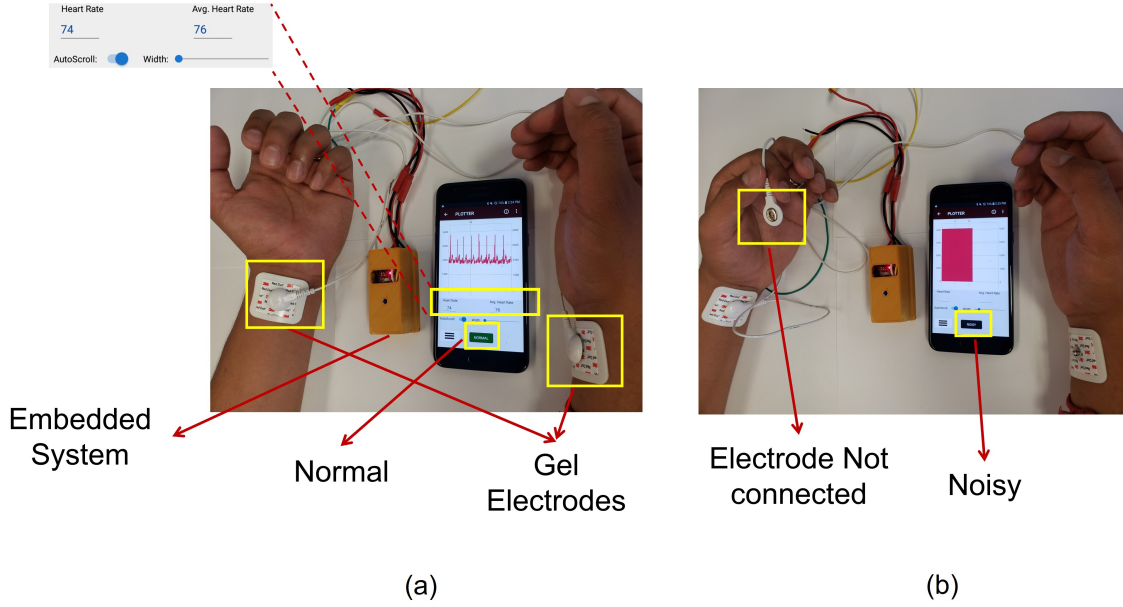


Fig. 4. Real-time data collection (a) Normal Signals (b) Noisy Signals.

establishes a BLE connection with the device and is ready to collect data. The user can then navigate to the *Modules* section, where they can view the *Data* or observe the ECG signals on the *Plotter*. Users can track their heart rate in *Plotter* section over an extended period of time in order to spot any potential problems. Additionally, the Smart-Health application provides users with an overview of their health through pre-trained model classification. In the *Plotter* section, there is a *Textbox* at the bottom where users can view their ECG signals' corresponding classification.

A depiction of the real-time data collection procedure is presented in Figure 4. The experiment involved attaching electrodes to the wrists of users and utilizing an embedded system to collect data. This data was then transmitted to the Smart Health application via Bluetooth connectivity. The application processed the data and provided real-time information on heart rate and various heart rhythm patterns. The objective was to evaluate the application's ability to accurately display and interpret users' heart rates and identify different types of heartbeats. From Figure 4(a), we can see the user's ECG signal is *Normal* and the average Heart Rate is 76bpm. By detaching the electrodes, artificial noise can be created, which results in a noisy signal (Figure 4(b)).

2.3 Pre-Trained Model

Feature extraction: In our research, utilizing the Time Series Feature Extraction Library (TS-FEL) in Python, we were able to extract a comprehensive set of 175 features from the analyzed beats. To select the most informative features for classification, we applied analysis of variance (ANOVA) algorithms. ANOVA, a statistical technique, is utilized to examine variances in mean values across different groups. This method aids in pinpointing the features that significantly influence the accurate classification of ECG beats.

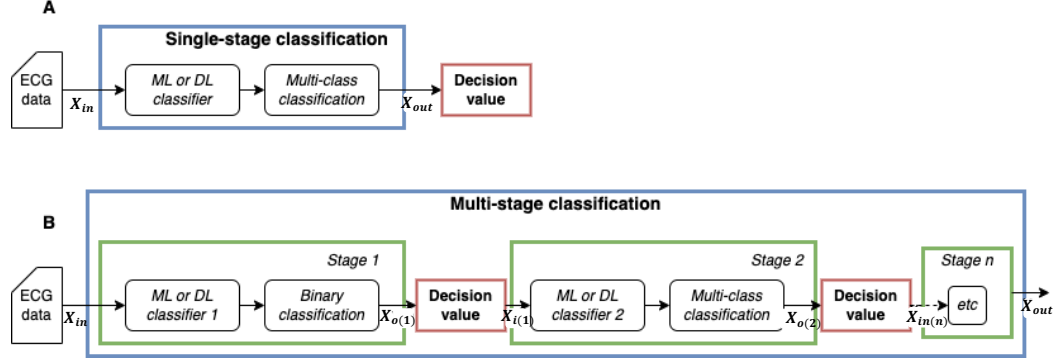


Fig. 5. The structure of single-stage and multi-stage classification is presented.

Classification: We performed heartbeat classification by assigning Normal heartbeats (N) as 0, Supraventricular Ectopic heartbeat (SV) as 1, Ventricular Ectopic beats (V) as 2, Fusion beats (F) as 3, and Unknown beats (Q) as 4. In our pursuit of precise classification, we harnessed a variety of machine learning methodologies, encompassing Decision Tree (DT), Artificial Neural Network (ANN), Support Vector Machine (SVM), Naive Bayes, and K-Nearest Neighbors (KNN), in addition to Bagged Tree. Furthermore, we utilized advanced Deep Learning models such as Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM). We tested both single-stage classification and two-stage classification methods to optimize the accuracy of the results. Figure 5 shows the structure of single and multi-stage classification.

Single-stage classification: To perform single-stage classification, we first evaluated the effectiveness of various machine learning and deep learning models. We aimed to select the best-performing model by training and testing them with 10-fold cross-validation. This allowed us to assess the model's ability to generalize to unseen data and avoided overfitting. We also adjusted the parameters of the models during the training process to optimize their performance. By doing so, we can determine the ideal combination of hyperparameters that results in the best performance for each model.

Multi-stage classification: In multi-stage classification, we first performed a binary classification to distinguish normal from abnormal noise. Then, a new classifier is built for the multi-class classification. We execute experimental trials using Decision Trees (DT) and Artificial Neural Networks (ANN), as these have proven to be the top-performing models among machine learning techniques.

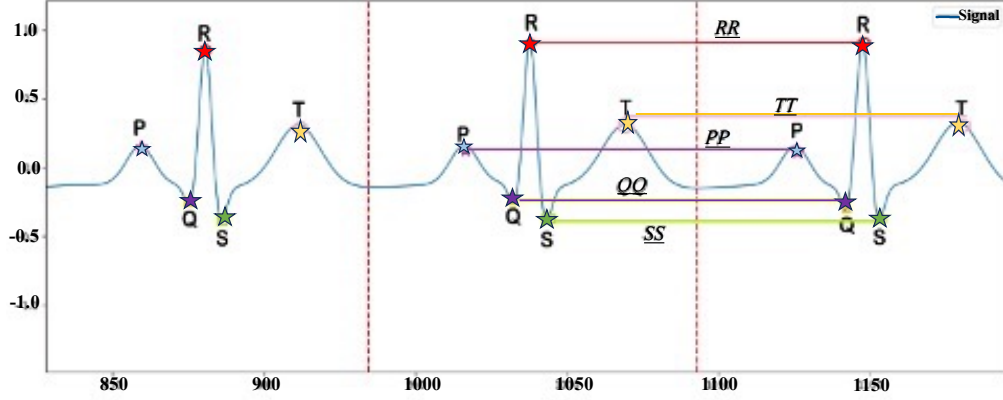


Fig. 6. The P, Q, R, S, and T peaks collectively create a heartbeat.

We also tweak parameters to gauge their performance. We also assessed power consumption, including memory usage, CPU usage, and running time to evaluate the efficiency of the classifiers.

Performance: To assess the effectiveness of our model, we employed a range of statistical metrics such as accuracy, precision, and recall. Furthermore, we evaluated the power consumption, including memory usage, CPU usage, and running time, of the top-performing machine learning (ML) and deep learning (DL) algorithms. The Keras Model Profiler, a Tensorflow package, was utilized to gather information on model parameters and memory requirements. To monitor system utilization, including CPU, memory, and network usage, we employed the Psutil package.

3 Results

3.1 Pre-Trained Model

Accurately detecting R peaks is crucial in ECG heartbeat recognition. Figure 6 illustrates the P, Q, R, S, and T peaks, as well as a heartbeat. We utilized the Pan-Tompkins algorithm to detect the precise position of the R peak, which in turn affects the accurate positioning of the P, Q, S, T, and T' peaks. The algorithm identifies the R peak by employing a sliding window that spans two heartbeats and advances one beat at a time until completion. The peaks detected by the algorithm are then labeled as P, Q, R, S, and T, as demonstrated in Figure 6, which showcases the interval from 850 to 1100 at the apex.

Table 1. Evaluate the performance of machine learning (ML) and deep learning (DL) methods on single-stage classification tasks by employing various classifier algorithms.

Single-stage Classification			
Classifier	Accuracy %	Precision %	Recall %
DT(10)	89	90	88
ANN(64)	94	92	93
SVM	78	78	79
Naive Bayes	81	83	82
KNN	75	75	74
Bagged tree	84	85	83
RNN	92	91	91
CNN	94	93	92
LSTM	90	89	87

In the machine learning domain, the Decision Tree (DT) classifier achieved an accuracy of 89%, with precision and recall rates of 90% and 88%, respectively. The Artificial Neural Network (ANN) exhibited an overall accuracy of 94%, accompanied by precision and recall scores of 92% and 93%. In the realm of deep learning, the Convolutional Neural Network (CNN) demonstrated superior performance, obtaining an accuracy of 94% and precision and recall values of 93% and 92%, respectively. Table 1 presents a comparative analysis of the performance of various machine learning (ML) and deep learning (DL) techniques in single-stage classification tasks.

Table 2 provides a summary of power consumption for single-stage classification using ANN and DT algorithms with varying parameters, such as the number of layers and maximum depth. For the ANN models, we explore performance across a range of layer counts, from 1 to 256, with 64 layers as the standard configuration. Both the 128-layer and 256-layer ANN classifiers achieved 100% accuracy without significant changes in memory, CPU usage, or runtime. Beyond 128 layers, the accuracy and power consumption decreased, while the runtime became faster. For the DT models, we analyzed performance by varying the maximum depth parameter from 1 to 25, with 10 as the standard setting. The classifiers with a maximum depth of 25 and 24 achieved 100% accuracy, again without noticeable changes in memory, CPU usage, or runtime. Beyond a maximum depth of 24, the accuracy and memory usage decreased, while CPU usage and runtime remained consistent.

Table 3 shows a summary of power consumption for multi-stage classification using ANN and DT algorithms with varying parameters and arrangements. For multi-stage ANN&ANN classifiers, we achieved 100% accuracy with 256 to 64 layers without much change in memory usage, but accuracy decreased after 64 layers in the first stage. In multi-stage DT&DT classifiers, we achieved 100% accuracy with 25 to 20 max depths without much change in memory usage in the first stage, and accuracy dropped off after 24 max depths in the second stage. Overall CPU usage and run times remained the same when max depths were reduced.

3.2 Application

The experiment involved validating the application’s functionality by simulating real-time data reception from the embedded system. Instead of using actual real-time data, we utilized an offline test dataset that contained labeled cardiac disease data. This allowed us to assess the application’s ability to accurately display the corresponding heart rhythms based on the provided labels.

Table 2. Power consumption for single-stage classification using ANN and DT algorithms with varying parameters.

Single-stage Classification				
Parameter	<i>Accuracy %</i>	<i>Memory usage (MiB)</i>	<i>CPU usage %</i>	<i>Run time(s)</i>
ANN(Layers)				
256	100	431	4	20
128	100	431	4	18
64	94	429	3.5	12
32	82	427	3.5	9
16	67	424	3.0	9
8	52	423	3.0	7
4	49	420	2.0	7
2	33	417	2.0	5
1	23	417	2.0	3
DT(MaxDepth)				
25	100	375	1.0	1
24	100	375	1.0	1
23	99	375	1.0	1
20	98	375	1.0	1
15	94	375	1.0	1
10	89	360	1.0	1
6	80	352	1.0	1
2	79	348	1.0	1
1	75	347	1.0	1

Table 3. Overview of the power consumption for multi-stage classification using ANN and DT algorithms with different parameters and arrangements.

Multi-stage Classification				
Parameter	Accuracy%	Memory usage (MiB)	CPU usage %	Run time(s)
	<i>First stage; Second stage</i>	<i>First stage; Second stage</i>		
ANN(Layers)				
256	100 ; 100	342 ; 389	17	42
128	100 ; 99	340 ; 387	16.5	35
64	100 ; 94	336 ; 387	7.5	20
32	98 ; 84	329 ; 375	14	15
16	95 ; 74	325 ; 368	13.7	11
8	87 ; 67	321 ; 365	13	11
4	74 ; 53	321 ; 357	12	10
2	68 ; 39	318 ; 357	11.5	7
1	67 ; 27	315 ; 351	16	5
DT(MaxDepth)				
25	100 ; 100	384 ; 375	1.0	1
24	100 ; 100	383 ; 375	1.0	1
23	100 ; 99	383 ; 374	1.0	1
20	100 ; 94	382 ; 374	1.0	1
15	98 ; 88	383 ; 375	1.0	1
10	95 ; 79	382 ; 375	1.0	1
6	91 ; 53	383 ; 374	1.0	1
2	83 ; 42	383 ; 374	1.0	1
1	74 ; 36	382 ; 373	1.0	1

As we don't have any offline dataset containing the Noisy signal, we couldn't validate our pre-trained model 1 which is a binary classifier and classifies ECG signals into Noisy and Not Noisy signals (fig 1). But it can detect the Noisy and Not Noisy signals in our application perfectly. We validated our pre-trained model 2 using the MIT-BIH test dataset. It contains 21892 samples and 18118 of them are Normal beats, 556 beats are Supraventricular Ectopic(SV), 1448 beats are Ventricular Ectopic(V), 160 bears are Fusion(F) and the rest 1610 beats are Unknown.

Table 4. Evaluate the performance of Artificial Neural Network (ANN) algorithm on Smart-Health Application using the offline dataset.

MIT-BIH test Dataset	Number of Samples	Accuracy %
Set 1	5000	93.30
Set 2	5000	92.66
Set 3	5000	93.14
Set 4	5000	92.86
Set 5	1892	83.93
Total	21892	91.178

We have separated the MIT-BIH test dataset into five distinct sets. Each set was passed to the Smart-Health application, which contained a pre-trained model designed to classify heart diseases. We used Artificial Neural Network (ANN) as a pre-trained model 2 because it exhibited an overall accuracy of 94% which showed the best performance among ML models. The pre-trained model classified each set based on the available ECG signals. After classification, the labels obtained from the Smart-Health application were compared with the MIT-BIH offline dataset. The accuracy was calculated for each set by comparing the obtained labels from the Smart-Health application with the ground truth labels from the offline dataset. The average accuracy across all five sets was computed to evaluate the performance of the pre-trained model on the MIT-BIH test dataset.

Table 4 provides the accuracy of five distinct sets over the application. The overall accuracy of the pre-trained ANN model was 91.178%.

The Smart-Health application detected 16955 Normal beats, 486 SV beats, 1247 V beats, 127 F beats and 1371 Unknown beats correctly. Figure 7 demonstrates the beats of the MIT-BIH test dataset and the corresponding accurate beats obtained from the Smart-Health application.

Arrhythmia refers to any abnormality in the rhythm of the heart's electrical activity. SVs, Vs, and fusion beats are all types of arrhythmias that can occur in the heart. Figure 8 shows some snapshots of the Smart-Health application after the detection of Normal, Noisy, and Arrhythmic beats. In some cases, arrhythmias can be serious and lead to heart failure, stroke, or sudden cardiac death.

4 Future Work

It is incredibly challenging to collect data when walking, jogging, or engaging in any other action because of the complex system setup. There are some IoT gadgets in the package that could be improved in the future, at which point we could quickly attach those devices to the body and collect data. We are developing a custom wearable ECG data collection hardware, that can significantly

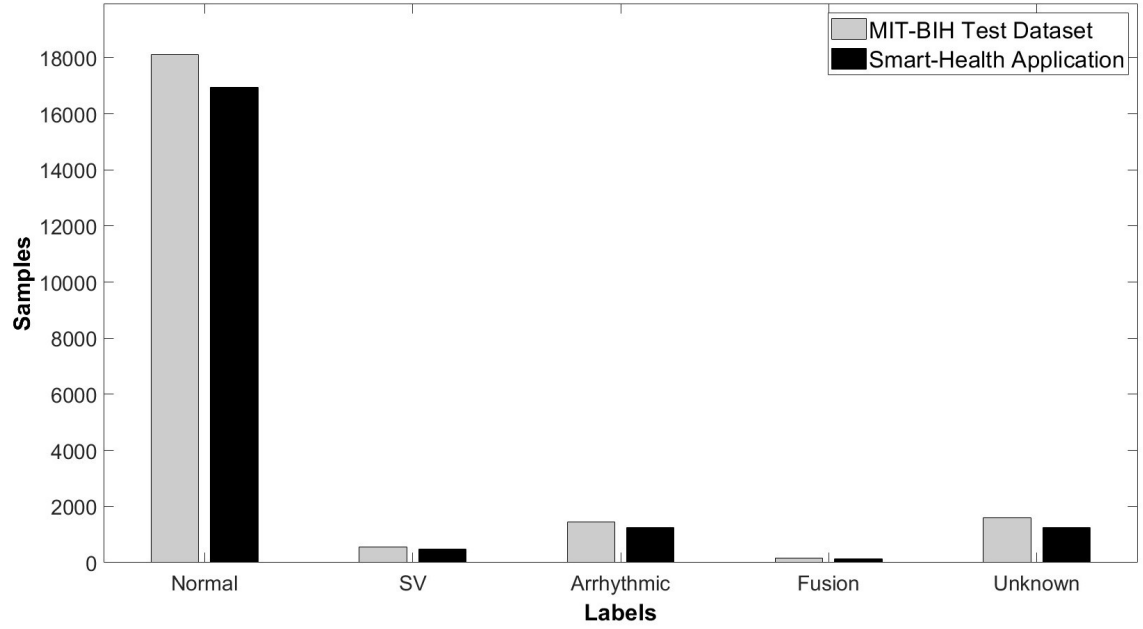


Fig. 7. Comparison of beat classification between the MIT-BIH test dataset and Smart-Health application.

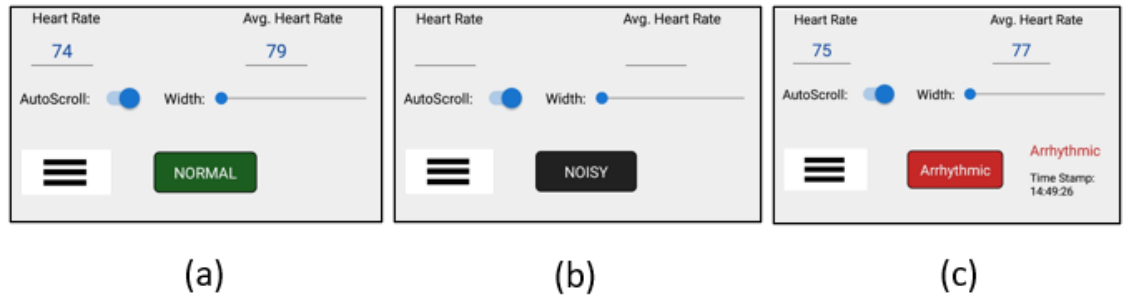


Fig. 8. Signal Detection (a) Normal (b) Noisy (c) Arrhythmic.

simplify the data collection process. Also, the accuracy of the ANN model on the application can be improved. We will also try other algorithms which had a lower accuracy on offline datasets but can perform well with real-time data.

At present, the Smart-Health application can detect some types of cardiovascular diseases, including Arrhythmia, and show ECG signals and instantaneous Heart Rate. As we have passed the MIT-BIH test dataset through the application to detect heart diseases and found quite a good accuracy, our next goals are:

- Plot study of the application with cardiac patients at a cardiac clinic.
- Improve our algorithms and find the best pre-trained model for the Smart-Health application to detect diseases.

5 Conclusion

In this study, we presented Smart-Health, a smartphone application that can continuously monitor ECG data, display Heart Rate and detect cardiac diseases using a pre-trained machine-learning model. The MIT-BIH test dataset was used to evaluate our model, and the findings suggest that our application can accurately detect various heart conditions. Patients can use this application to check their heart health in real time and take appropriate steps depending on the results. Overall, the Smart-Health application has the potential to be a valuable tool for the early detection and monitoring of cardiac problems. The Smart-Health application can aid in the prevention and control of cardiovascular illnesses, resulting in better health outcomes and a higher quality of life for patients by empowering them to actively participate in their own health management.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 2105766. The development of the ECG device was performed by Mahfuzur Rahman, Robert Hewitt, and Bashir I. Morshed.

References

1. Glovaci D, Fan W, Wong ND. Epidemiology of diabetes mellitus and cardiovascular disease. *Current cardiology reports*. 2019 Apr;21:1-8.
2. Walker A, Muhlestein J. Smartphone electrocardiogram monitoring: current perspectives. *Advanced Health Care Technologies*. 2018 Jun 8;4:15-24.
3. Koltowski L, Balsam P, Glowczynska R, Rokicki JK, Peller M, Maksym J, Blicharz L, Maciejewski K, Niedziela M, Opolski G, Grabowski M. Kardia Mobile applicability in clinical practice: A comparison of Kardia Mobile and standard 12-lead electrocardiogram records in 100 consecutive patients of a tertiary cardiovascular care center. *Cardiology journal*. 2021;28(4):543-8.
4. William AD, Kanbour M, Callahan T, Bhargava M, Varma N, Rickard J, Saliba W, Wolski K, Hussein A, Lindsay BD, Wazni OM. Assessing the accuracy of an automated atrial fibrillation detection algorithm using smartphone technology: The iREAD Study. *Heart rhythm*. 2018 Oct 1;15(10):1561-5.
5. Haverkamp HT, Fosse SO, Schuster P. Accuracy and usability of single-lead ECG from smartphones-A clinical study. *Indian pacing and electrophysiology journal*. 2019 Jul 1;19(4):145-9.

6. Raja JM, Elsagr C, Roman S, Cave B, Pour-Ghaz I, Nanda A, Maturana M, Khouzam RN. Apple watch, wearables, and heart rhythm: where do we stand?. *Annals of translational medicine*. 2019 Sep;7(17).
7. Rahman MJ, Morshed BI, Harmon B, Rahman M. A pilot study towards a smart-health framework to collect and analyze biomarkers with low-cost and flexible wearables. *Smart Health*. 2022 Mar 1;23:100249.
8. T. E. Senator, "Multi-stage classification," Fifth IEEE International Conference on Data Mining (ICDM'05), Houston, TX, 2005, pp. 8 pp.-, doi: 10.1109/ICDM.2005.102.
9. Abawajy JH, Kelarev AV, Chowdhury M. Multistage approach for clustering and classification of ECG data. *Computer methods and programs in biomedicine*. 2013 Dec 1;112(3):720-30.
10. Hosseini HG, Reynolds KJ, Powers D. A multi-stage neural network classifier for ECG events. In 2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society 2001 Oct 25 (Vol. 2, pp. 1672-1675). IEEE.
11. Kutlu Y, Altan G, Allahverdi N. Arrhythmia classification using waveform ECG signals. In Int. Conf. Advanced Technology & Sciences, Konya, Turkey 2016.
12. Altan G, Kutlu Y, Allahverdi N. A multistage deep belief networks application on arrhythmia classification. *International Journal of Intelligent Systems and Applications in Engineering*. 2016 Dec 26;4(Special Issue-1):222-8.
13. Xiaolin L, Panicker RC, Cardiff B, John D. Multistage pruning of CNN based ECG classifiers for edge devices. In 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) 2021 Nov 1 (pp. 1965-1968). IEEE.
14. Zheng J, Chu H, Struppa D, Zhang J, Yacoub SM, El-Askary H, Chang A, Ehwerhemuepha L, Abudayyeh I, Barrett A, Fu G. Optimal multi-stage arrhythmia classification approach. *Scientific reports*. 2020 Feb 19;10(1):2898.
15. Satija U, Ramkumar B, Manikandan MS. A simple method for detection and classification of ECG noises for wearable ECG monitoring devices. In 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN) 2015 Feb 19 (pp. 164-169). IEEE.
16. ANSI/AAMI. Testing and Reporting Performance Results of Cardiac Rhythm and ST Segment Measurement Algorithms, 1998. Association for the Advancement of Medical Instrumentation.
17. Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* 101(23):e215-e220 [Circulation Electronic Pages; <http://circ.ahajournals.org/content/101/23/e215.full>]; 2000 (June 13).
18. Mahfuzur Rahman, Robert Hewitt, and Bashir I. Morshed, Design and Packaging of a Custom Single-lead Electrocardiogram (ECG) Sensor Embedded with Wireless Transmission, IEEE Dallas Circuits and Systems (DCAS), 14-16 Apr. 2023.
19. Golden DP, Wolthuis RA, Hoffer GW. A spectral analysis of the normal resting electrocardiogram. *IEEE Transactions on Biomedical Engineering*. 1973 Sep(5):366-72.