CS6611 – CREATIVE AND INNOVATIVE PROJECT

Team no: 30

Mentor: Dr.R.Kathiroli

Team Members:

Aishwarya Devi R (2021503501)

Surya Raj M (2021503701)

Sridhar P (2021503709)

IoT-Enabled Non-Invasive Methods for Early Detection of Cardiotoxicity

Problem Statement:

The current state of cardiovascular health monitoring faces challenges in efficiently detecting cardio-toxicity. Accurate prediction of Cardiotoxicity remains a complex task, underlining the necessity for a portable device incorporating Deep Learning (DL) algorithms for improved diagnostics. This project proposes a non-invasive approach for cardiovascular health monitoring by implementing a real-time DL-based system that improves early detection, particularly in cases of cardio-toxicity, thereby enhancing overall patient health.

Objectives:

- To detect Arrythmia from the abnormalities in ECG.
- To develop a non-invasive tool using IoT-based cardiac biosensors and Deep Learning (DL) algorithms to predict cTn levels.
- To enable real-time predictions of heart functions for timely interventions and improved healthcare outcomes.
- To integrate Explainable Artificial Intelligence (XAI) techniques for interpretation and explainability of the model's predictions.
- To integrate the trained model into an IoT wearable device for continuous monitoring and early detection of cardiovascular abnormalities.

Literature Survey:

In [1], a comparative analysis was conducted on the classification of heart failure using 13 features, employing various models including CNN, RNN, MLP, and LSTM, with CNN demonstrating superiority with an accuracy of 92.89%. The study incorporated IoT sensors for monitoring heart rate, blood pressure (BP), temperature, blood glucose, cholesterol, and ECG signals. Additionally, transfer learning approaches, VGG16 and AlexNet, were evaluated alongside their training times. The validation was performed using

10-fold cross-validation. Limitations include, dataset is highly imbalanced hence model is not trained well on minority classes.

In [2], this paper proposes methodology for classifying cardiac arrythmia into 17 classes, considering long duration ECG signal of 10s. It employs 1D CNN with 16 layer deep, achievingaccuracy of 91.33% and classification time of 0.015 s. It particularly focuses on reducing the computational complexity. Limitations include small no of ECG signal fragments are analysed, no possibility of classifying fragments of ECG signal containing more than one class.

In [3], the study employs a Modified Deep CNN for classification and integrates the Elephant Herd Optimization Algorithm (AEHO), while feature selection is done by Cuttlefish Optimization Algorithm. Simulation is achieved through the integration of microcontroller and LoRa communication hardware for data transmission to the cloud, incorporating the Omron HeartGuide smartwatch for blood pressure measurement and the AD8232 for ECG measurement. Limitations include actual data for serum cholestrol, chest pain and glucose level are not used, pseudo numbers are generated.

In [4], the study employs single-lead ECG to detect cardiotoxicity in cancer patients with minimal cardiovascular diseases following the first cycle of polychemotherapy, utilizing the CardioQvark mobile phone cover with an integrated ECG sensor for signal detection. Detection focuses on identifying left ventricular diastolic dysfunction, atrial fibrillation, and QTc prolongation. The limitations are sample did not include patients with intermediate and low LVEF.

In [5], the proposed wearable ECG monitoring system, IREALCARE, consists of an integrated ECG sensor where ECG data is collected, transmitted to the control unit, sampled at 250Hz, converted to digital signals using ADC, and wirelessly transmitted to a mobile device via Bluetooth. Preprocessing of the ECG involves enlarging by sliding window and denoising using Discrete Wavelet Transform (DWT). The system employs a confidence level-based training approach with ResNet as the training model, achieving an accuracy of 90.2%. Limitations include high reliability on the optimal selection of the confidence level, which may vary based on the dataset and label quality.

In [6], the approach utilized in the study is LSTM-DBN, achieving an accuracy of 88.42%. This approach combines LSTM for learning long-term dependencies with DBN for feature selection. Stochastic Gradient Descent (SGD) is employed for optimizing the loss function. From four datasets, twelve main features are extracted, including Heart Rate Variability (HRV) from ECG signals, used for classifying cardiovascular diseases. Comparative analysis is conducted with four other deep learning approaches (CNN, RNN, GRU, Ensemble) and four machine learning approaches (MLP, LR, SVM, RF). Limitations include high computational efficiency and memory is required for DBN as compared to other algorithms.

In [7], the study focuses on remote monitoring of cardiac signals through the integration of IoT and machine learning techniques, with detction of various arrhythmia including ventricular, supraventricular, and fusion beats. The components include the Polar H10 ECG sensor for signal acquisition, a CRUD REST API for data management, a Monitoring GUI for visualization, an Arrhythmia Detection Component for automated

analysis, all facilitated by the MQTT protocol for communication. The chosen algorithm for arrhythmia classification is k-nearest neighbors with accuracy of 97%. Limitations include lack of discussion on robustness to noise and variability in ECG signals.

In [8], the system enables real-time monitoring of ECG signals by filtering and localizing R-peaks using the Pan Tompkin's algorithm. Feature extraction is performed using a combination of Fast Fourier Transform (FFT) and discrete wavelet transform (DWT). Classification of extracted features is achieved through twin support vector machines (TSVM) tuned by Particle Swarm Optimization (PSO). The prototype is implemented on a microcontroller platform equipped with a Wi-Fi module for data transmission and connectivity. Limitations include more specific selection of features based on R-wave morphology, lacks the discussion about power consumption of the platform.

In [9], the approach utilized is ECG-GAN for data augmentation and the ResNet BiLSTM-Attention for classification achieving accuracy of 99.4%. ResNet is used for local feature extraction and BiLSTM for global feature extraction. DWT is used for denoising and QRS waveform detected using Pan Tompkin's algorithm. Limitations include, integrating ResNet and BiLSTM models introduces complexity to the classification process, optimization of hyperparameters is not performed.

In [10], this paper presents a lightweight Convolutional Neural Network (CNN) model for predicting four major cardiac abnormalities using a public dataset of ECG images. The study investigates transfer learning using low-scale pretrained deep neural networks (SqueezeNet and AlexNet) and proposes a new CNN architecture for cardiac abnormality prediction achieving 99.79% accuracy. The pretrained models and the proposed CNN model are used as feature extraction tools for traditional machine learning algorithms. Limitations include, training and testing time for SquuezeNet based algorithm is longer, optimization algorithms are not used to determine the value of hyperparameters.

Contributions:

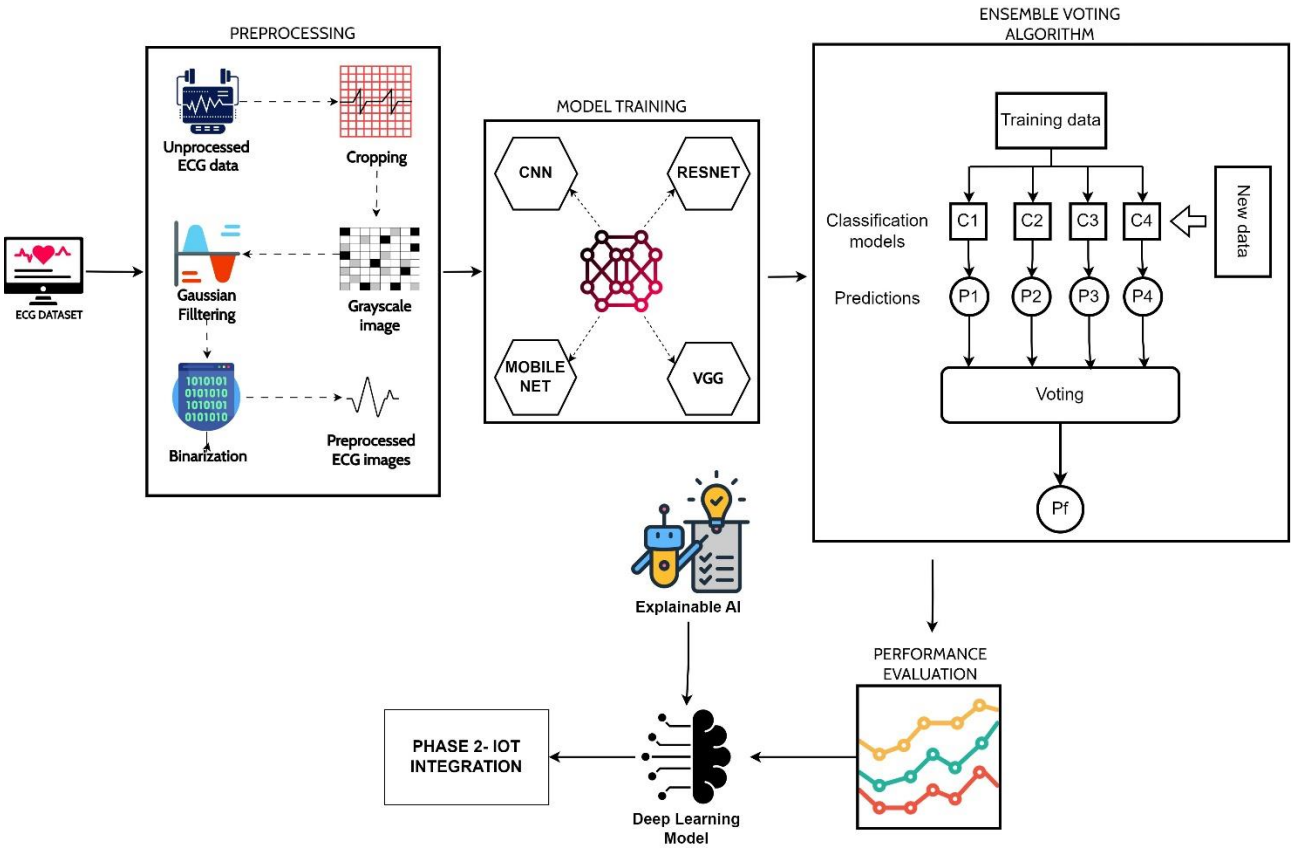
- **Ensemble Model:** The project evaluates four different classification models - CNN, MobileNet, ResNet, and VGG - to determine the most suitable ones for predicting heart failure based on the integrated data. Each model is assessed based on its accuracy, sensitivity, specificity, and computational efficiency. The ensemble approach combines the strengths of these individual models to enhance predictive performance, leveraging their diverse capabilities to achieve better overall results.

- **Early Detection of Cardiotoxicity:** By monitoring cTn levels, the platform enables early detection of cardiac damage, allowing for timely intervention and potentially preventing serious complications. This is particularly important for cancer patients undergoing treatment that may impact cardiovascular health.

- **Non-Invasive Monitoring:** The platform offers a non-invasive method for measuring HF, reducing the need for invasive procedures and improving patient comfort and compliance. This makes it suitable for long-term monitoring and management of cardiac health.

- **Potential for Scalability and Accessibility:** The proposed platform has the potential for scalability and accessibility, making it suitable for widespread deployment in diverse healthcare environment.

Architecture diagram:



Modules:

1. Preprocessing of ECG images
2. Model training using CNN
3. Model training using MobileNet
4. Model training using VGG
5. Model training using ResNet
6. Ensemble voting algorithm
7. Integration of Explainable AI using Grad CAM
8. IoT device integration

Algorithm:

a. Preprocessing Algorithm:

Input: 12 lead ECG image (input_image)

Output: Preprocessed ECG image

Procedure:

1. Crop the 12-lead ECG image
   cropped_image = input_image.crop(x_{left}, x_{top}, x_{right}, x_{bottom})
2. Convert ECG image to grayscale
   grayscaled_image = rgb2gray(cropped_image)
3. apply Gaussian blur with sigma value 0.7
   blurred_image=gaussian(grayscaled_image, σ=0.7)
4. apply Otsu thresholding
   threshold = threshold_otsu(blurred_image)
   binary_image = blurred_image < threshold
5. Return the binary image after thresholding

b. MobileNet Algorithm:

Input: Preprocessed ECG image (X, y), Number of classes (num_classes), Number of epochs (num_epochs), Batch size (batch_size), Learning rate (learning_rate)

Output: Trained MobileNet model

Procedure:

1. Import necessary modules
2. Split the input data into train and test sets
   X_{train}, X_{test}, y_{train}, y_{test} = train_test_split(X, y, test_size=0.2)
3. Setup the pipeline steps:
   ('conv1', Conv2D(64, (3, 3)))
   ('batch_norm1', BatchNormalization())
   ('activation1', Activation('relu'))
   ('depthwise_conv1', DepthwiseConv2D((3, 3)))
   ('batch_norm2', BatchNormalization())
   ('activation2', Activation('relu'))
   ('conv2', Conv2D(128, (1, 1)))
   ('max_pooling', MaxPooling2D((2, 2)))
   ('global_avg_pooling', GlobalAveragePooling2D())
   ('dropout', Dropout(0.5))
   ('output', Dense(len(classes), activation='softmax'))
4. Compile the model using Adam optimizer
   optimizer = Adam(learning_rate)

model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])

5. Define callbacks for model training:
   checkpoint = ModelCheckpoint("best_model.h5", monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
   reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=3, min_lr=0.0001, verbose=1)
   early_stop = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
6. Train the model:
   history = model.fit(X_{train}, y_{train}, num_epochs, batch_size, validation_data=(X_{test}, y_{test}), callbacks=[checkpoint, reduce_lr, early_stop])
7. Predict the labels of the test set: *y_pred = cv.predict(X_{test})*
8. Compute accuracy: *accuracy = accuracy_score(y_{test}, y_pred)*
9. Generate accuracy and classification report

c. CNN Algorithm

Input: Preprocessed ECG image (X, y), Number of classes (num_classes), Number of epochs (num_epochs), Batch size (batch_size), Learning rate (learning_rate)

Output: Trained CNN model

Procedure:

1. Import necessary modules
2. Split the input data into train and test sets
   X_{train}, X_{test}, y_{train}, y_{test} = train_test_split(X, y, test_size=0.2)
3. Setup the pipeline steps
   ('conv1', Conv2D(32, (3, 3), input_shape=(64, 64, 3)))
   ('activation1', Activation('relu'))
   ('max_pooling1', MaxPooling2D(pool_size=(2, 2)))
   ('conv2', Conv2D(64, (3, 3)))
   ('activation2', Activation('relu'))
   ('max_pooling2', MaxPooling2D(pool_size=(2, 2)))
   ('flatten', Flatten())
   ('dense1', Dense(128))
   ('activation3', Activation('relu'))
   ('dropout', Dropout(0.5))
   ('dense2', Dense(num_classes))
   ('output', Activation('softmax'))
4. Compile the model using Adam optimizer
   optimizer = Adam(learning_rate)
   model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])

5. Define callbacks for model training:
   checkpoint = ModelCheckpoint("best_model.h5", monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
   reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=3, min_lr=0.0001, verbose=1)
   early_stop = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
6. Train the model:
   history = model.fit(X_{train}, y_{train}, num_epochs, batch_size, validation_data=(X_{test}, y_{test}), callbacks=[checkpoint, reduce_lr, early_stop])
7. Predict the labels of the test set: *y_pred = cv.predict(X_{test})*
8. Compute accuracy: *accuracy = accuracy_score(y_{test}, y_pred)*
9. Generate accuracy and classification report

d. Ensemble Voting Algorithm

Input: Trained Models: $model_1$, $model_2$, $model_3$, $model_4$
Output: Ensemble Prediction (*ensemble_pred*)
Procedure:
1. Initialize *ensemble_pred* = []
2. For each sample *i* =1 to *n*:
   a. Initialize *class_votes*
   b. For *j* = 1 to 4:
      i. *pred = $model_j$.predict(sample_data[i])*
      ii. If *pred* not in *class_votes:*
          class_votes [pred] = 1
      iii. Else:
          class_votes [pred]++
   c. If there is a tie:
      i. Initialize *weighted_votes = 0*
      ii. For each model *k* with tied predictions:
      iii. Calculate the weight w_k based on the model's performance on the validation set as $w_k = \frac{accuracy_k}{\sum_{i=1}^{N} accuracy_i}$
      iv. Calculate weighted vote for model *k* as *weighted_vote$_k$ = w_k * class_votes [prediction_by_model]*
      v. Determine the prediction with the highest weighted sum: *ensemble_pred[i] = argmax (weighted_votes)*
   d. Else (no tie):
      i. *ensemble pred[i]= argmax (class_votes)*
3. Return *ensemble_pred*

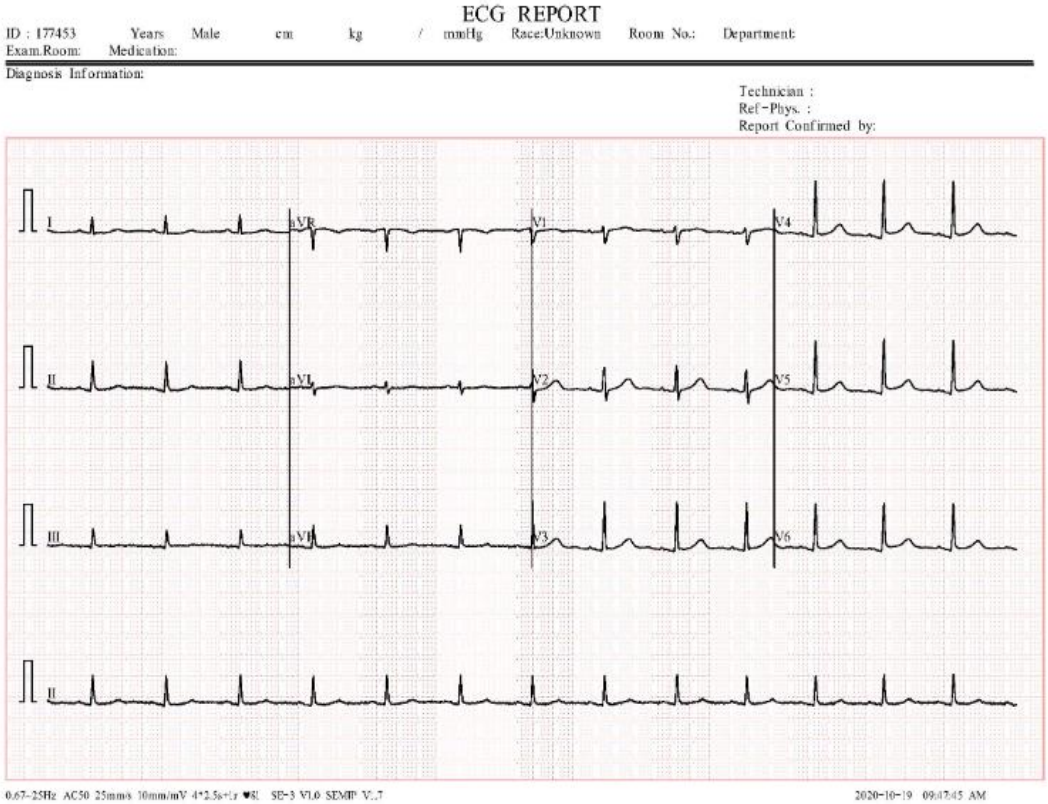
e. Visualization using Grad CAM

Input: Preprocessed input image Trained CNN model Target class index
Output: Grad-CAM heatmap *H*
Procedure:

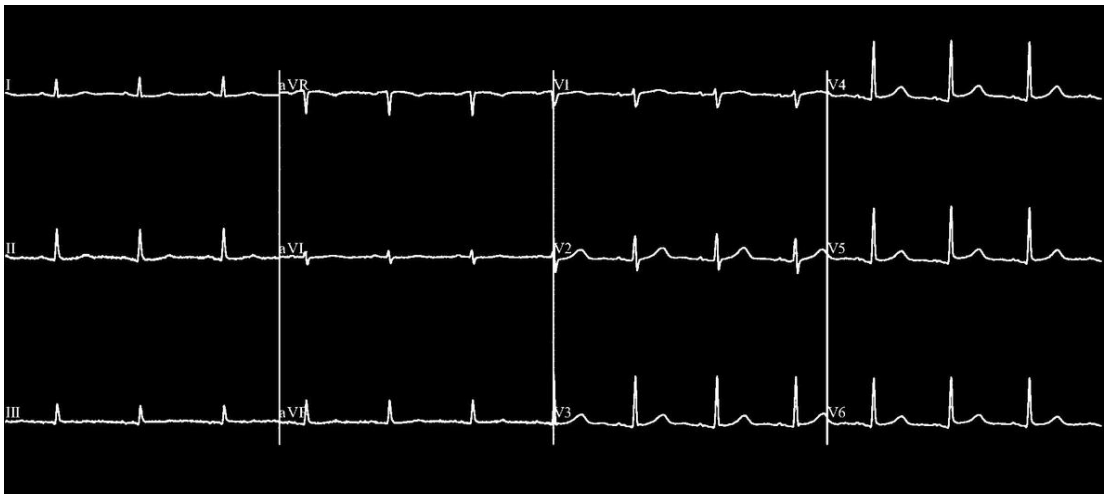
1. Load the preprocessed input image *I*
2. Pass the input image through the model M to obtain the feature maps: *F = M(I)*
3. Retrieve the final convolutional layer of the model *M* and denote it as *A*.
4. Compute the gradient of the target class score with respect to the feature maps of the final convolutional layer: $\partial L/\partial Aj$ where L is the target class score and A_j are the activations in the final convolutional layer.
5. Compute the importance weights using global average pooling (GAP) of the gradients along the spatial dimensions:$w_k = \sum_{i=1}^{H}\sum_{j=1}^{W} \partial L/\partial Ai,j$
   where *H* and *W* are the height and width of the feature maps, respectively.
6. Compute the Grad-CAM heatmap $H = \sum_k W_k F_k$
7. Apply ReLU activation to the Grad-CAM heatmap: *H = ReLU(H)*
8. Resize the Grad-CAM heatmap to match the size of the input image: *H = Resize(H, input_size)*
9. Normalize the Grad-CAM heatmap to the range [0, 1] by min-max scaling: *H-min(H)/ max(H)-min(H)*
10. Visualize the Grad-CAM heatmap for interpretation.

Implementation:

Unprocessed 12 lead ECG image:



Preprocessed ECG image



Model training using CNN

```
19/19 [==============================] - 12s 650ms/step - loss: 0.2257 - accuracy: 0.9209 - val_loss: 0.3044
Epoch 31/50
19/19 [==============================] - ETA: 0s - loss: 0.2336 - accuracy: 0.9259
Epoch 31: val_accuracy did not improve from 0.91946
19/19 [==============================] - 12s 660ms/step - loss: 0.2336 - accuracy: 0.9259 - val_loss: 0.2861
5/5 [==============================] - 2s 373ms/step - loss: 0.2704 - accuracy: 0.9128
Test Loss: 0.2703878879547119
Test Accuracy: 0.9127516746520996
```

Model training using MobileNet

```
19/19 [==============================] - 44s 2s/step - loss: 0.0894 - accuracy: 0.9753 - val_loss: 0.0583
Epoch 42/50
19/19 [==============================] - ETA: 0s - loss: 0.0900 - accuracy: 0.9703
Epoch 42: val_accuracy did not improve from 0.99329
19/19 [==============================] - 47s 2s/step - loss: 0.0900 - accuracy: 0.9703 - val_loss: 0.0926
5/5 [==============================] - 9s 2s/step - loss: 0.0705 - accuracy: 0.9866
Test Loss: 0.07050875574350357
Test Accuracy: 0.9865771532058716
```

Evaluation Metrics:

1. Accuracy measures the proportion of correctly classified samples among the total number of samples.

   $$\text{Accuracy} = \text{TP} + \text{TN}/\text{TP} + \text{TN} + \text{FP} + \text{FN}$$

2. Precision measures the proportion of true positive predictions among all positive predictions.

   $$\text{Precision} = TP/TP + FP$$

3. Recall measures the proportion of true positive predictions among all actual positive samples.

Recall = $TP/TP + FN$

4. Specificity measures the proportion of true negative predictions among all actual negative samples.

   Specificity = $TN/TN + FP$

5. F1-score measures the reliability of the model by calculating the harmonic mean of its precision and recall

   F1-score = $2 \times \text{Precision} \times \text{Recall} / Precision + Recall$

6. AUROC curve provides an aggregate measure of performance across all possible classification thresholds.
7. Confusion matrix is a table that represents the counts of true positive, true negative, false positive, and false negative predictions.

References:

1. Umer M, Sadiq S, Karamti H, Karamti W, Majeed R, Nappi M. IoT Based Smart Monitoring of Patients' with Acute Heart Failure. Sensors (Basel). 2022 Mar 22;22(7):2431. doi: 10.3390/s22072431. PMID: 35408045; PMCID: PMC9003513.

2. Yıldırım Ö, Pławiak P, Tan RS, Acharya UR. Arrhythmia detection using deep convolutional neural network with long duration ECG signals. Comput Biol Med. 2018 Nov 1;102:411-420. doi: 10.1016/j.compbiomed.2018.09.009. Epub 2018 Sep 15. PMID: 30245122.

3. M. A. Khan, "An IoT Framework for Heart Disease Prediction Based on MDCNN Classifier," in *IEEE Access*, vol. 8, pp. 34717-34727, 2020, doi: 10.1109/ACCESS.2020.2974687.

4. Mesitskaya DF, Fashafsha ZZA, Poltavskaya MG, Andreev DA, Levshina AR, Sultygova EA, Gognieva D, Chomakhidze P, Kuznetsova N, Suvorov A, Marina I S, Poddubskaya E, Novikova A, Bykova A, Kopylov P. A single-lead ECG based cardiotoxicity detection in patients on polychemotherapy. Int J Cardiol Heart Vasc. 2024 Jan 20;50:101336. doi: 10.1016/j.ijcha.2024.101336. PMID: 38304727; PMCID: PMC10831811.

5. Wang, Peng & Lin, Zihuai & Yan, Xucun & Chen, Zijiao & Ding, Ming & Song, Yang & Meng, Lu. (2022). A Wearable ECG Monitor for Deep Learning Based Real-Time Cardiovascular Disease Detection. arXiv:2201.10083.

6. Dami, S., Yahaghizadeh, M. Predicting cardiovascular events with deep learning approach in the context of the internet of things. Neural Comput & Applic 33, 7979–7996 (2021).

7. Cañón-Clavijo RE, Montenegro-Marin CE, Gaona-Garcia PA, Ortiz-Guzmán J. IoT Based System for Heart Monitoring and Arrhythmia Detection Using Machine

Learning. J Healthc Eng. 2023 Feb 8;2023:6401673. doi: 10.1155/2023/6401673. PMID: 36818385; PMCID: PMC9931473.

8. S. Raj, "An Efficient IoT-Based Platform for Remote Real-Time Cardiac Activity Monitoring," in *IEEE Transactions on Consumer Electronics*, vol. 66, no. 2, pp. 106-114, May 2020, doi: 10.1109/TCE.2020.2981511.

9. Ma S, Cui J, Xiao W, Liu L. Deep Learning-Based Data Augmentation and Model Fusion for Automatic Arrhythmia Identification and Classification Algorithms. Comput Intell Neurosci. 2022 Aug 11;2022:1577778. doi: 10.1155/2022/1577778. PMID: 35990162; PMCID: PMC9388256.

10. M. B. Abubaker and B. Babayiğit, "Detection of Cardiovascular Diseases in ECG Images Using Machine Learning and Deep Learning Methods," in *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 2, pp. 373-382, April 2023, doi: 10.1109/TAI.2022.3159505.