Predictive ECG Monitoring with Machine Learning: Detecting Sudden Cardiac Arrest and Assessing Hospitalization Risk

MSc Research Project
MSc Data Analytics

Dona Joy Alappatt
Student ID: X23276657

School of Computing
National College of Ireland

Supervisor: Christian Horn

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: …Dona Joy Alappatt…………………………………………………………………

Student ID: …x23276657……………………………………………………………………………..……

Programme: … MSc Data Analytics ……………………………… **Year:** ……2024-2025..

Module: …MSc Research Practicum ……………………………………………………..………

Supervisor: … Christian Horn ………………………………………………………………..………

Submission Due Date: ……01-09-2025……………………………………………………………………………..………

Project Title: Predictive ECG Monitoring with Machine Learning: Detecting Sudden Cardiac Arrest and Assessing Hospitalization Risk

Word Count: …1733……………………… **Page Count**………5……………………………………..……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: …Dona Joy Alappatt…………………………………………………

Date: …31-08-2025……………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Dona Joy Alappatt
Student ID: X23276657

1 Introduction

Sudden Cardiac Arrest (SCA) is a significant global health problem, accounting for 15-20% of worldwide deaths. Despite the implementation of improved first response systems, the rates of survival remain exceedingly low. The challenge is in precisely predicting and preventing sudden cardiac arrest (SCA). ML algorithms provide an accurate and automated alternative to traditional methods for identifying and diagnosing cardiac arrest. These technologies employ advanced algorithms to analyze vast quantities of medical data and identify patterns that indicate cardiac arrest. The utilization of artificial intelligence (AI) and machine learning (ML) in the healthcare sector is expanding, with evident applications in medical diagnostics, statistics, and precision medicine. The objective of this study is to create a method for continuously monitoring ECG data in real-time in order to identify instances of sudden cardiac arrest to find if the patient is going to be hospitalised. This config file will help in running the entire code in steps and help in understanding the code process.

2 Systems Requirements

OS: Windows 11H2, Ubuntu 16.04
RAM: 12 GB
Graphics: Nvidia RTX 3010, 12 GB
Processor: Intern i9 13th Gen

3 Code Explanation

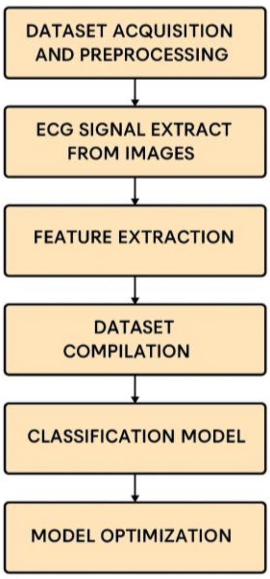

Figure 1: Algorithm Flow

There are two codes that are made based on the type of the datasets,
Case 1: Signal Data (2 datasets are used)–
Data:
https://drive.google.com/drive/u/2/folders/1PzkHw59DQMxsjjyeZcm_lUyvxmbrBk-y
https://drive.google.com/drive/u/2/folders/1EGg6_ZWlBqbC_klCRA8tr6Ooz4cpwV2B

Code: https://drive.google.com/drive/u/2/folders/1YYw9Hb8fiilVDuA4xq_OFRb9vQjfe559

Case 2: Image Data –
Data:
https://www.kaggle.com/datasets/erhmrai/ecg-image-data
Kaggle json file used for this uploading to the colab –
https://drive.google.com/drive/u/2/my-drive

Code: https://drive.google.com/drive/u/2/folders/1tQ3oYcRJyVTzq633PIqecZQ1d4Mn1Vdl

4 Algorithm Flow Descriptions

Step 1: Dataset Acquisition and Preprocessing – A collection of electrocardiogram (ECG) pictures, belonging to the groups of myocardial infarction, normal, and abnormal heartbeat, was obtained on Kaggle in the present stage. The photos were organized into easily navigable directory names. In order to make all of the photos the same size and accommodate the remaining data flow, they were downscaled to 224 × 224 pixels and converted into tensors using the PyTorch torchvision transform. To ensure the dataset's integrity, a visual inspection of each category's photographs was conducted.
Step 2: ECG Signal Extraction from Images – Each ECG was examined in order to separate the signal waveform. After that, the images were converted to grayscale. After flipping them, a threshold was added to make the waveform clearly distinct. OpenCV was used to locate the traced contour, and the ECG signal was transformed into a one-dimensional sequence by averaging the pixel values along each vertical column. The resultant signal was prepared for feature extraction by making it zero-mean and unit-variance ready.
Step 3: Feature Extraction – A good set of characteristics was produced by extracting features from digital electrocardiogram (ECG) signals. Amplitude, peak count, RR intervals, heart rate, slope statistics, and mean ST-segment were among the time-domain features. The fast Fourier transform (FFT), which incorporates dominant frequency, total power, spectral entropy, and centroid measurement, was used to analyze the frequency-domain spectrum. Morphological parameters such the PR, QRS, QT, P- and T-wave amplitudes, and ST deviation were among them. Complementary metrics derived from image analysis, such as the number of contours, edge density, and intensity distribution, were included.
Step 4: Dataset Compilation – Every feature that was taken from the electrocardiogram (ECG) pictures was collected and stored in databases. Consequently, the extracted features and the class labels were included in comma-separated value (CSV) files for training and testing data. These datasets were utilized to train the machines on the next machine-learning models.
Step 5: Classification Model – A random forest classifier model was trained with the collected data to categorize ECG pictures into the appropriate groups. After being trained on the training set's features, the model was applied to the test and validation sets. Class-wise performance was examined by generating confusion matrices and calculating positives like accuracy, precision, and recall.

Step 6: Model Optimization – To maximize the model's performance, manual hyperparameter adjustment was used rather than machine learning. A search in various parameters was conducted by systematically altering max_depth and n_estimators. To ensure a proportionate number of occurrences of each class, this data was divided into training and validation sets. The goal of this procedure was to determine the classifier's optimal configuration in terms of accuracy and generalizability.

5 Steps to Reproduce

Step 1: Log into the Drive

Step 2: Authenticate the drive access by selecting the required cell that helps in authenticating

Step 3: Run all

6 Code Contribution

By means of the introduction of a systematic machine learning strategy, the code that was created in this project offers direct assistance in the construction of the artifact that identifies Sudden Cardiac Arrest (SCA) based on the observation of electrocardiograms:

1. **Acquisition of Data**

Following the reading of electrocardiogram (ECG) signals in CSV format, the code cleans and preprocesses the data and prepares it to be fed to machine learning pipelines. This applies to datasets which are signal based. Image-based datasets In picture-based datasets, the code gains access to the Kaggle ECG image dataset by feeding a file of a Kaggle JSON API key to Google Colab. By this direct connectivity, datasets can be obtained without any interruptions and synchronized to the training environment.

2. **Preprocessing and Extracting Signals.**

On execution of the code, ECG inputs are normalized, pictures are downsampled and tensors are turned into numbers. To achieve consistency in the datasets, preprocessing tools relying on OpenCV and PyTorch extract the waveforms on electrocardiogram (ECG) pictures and transform them into 1D sequences that can be utilized.

3. **Features Engineering**

This process derives both time-domain features (heart rate, intervals and amplitudes) and frequency-domain features (FFT-based features) and morphological features (wave patterns and contour density). This is then tabulated in structured CSV files to be used in training and evaluation.

4. **Training and Evaluation of Models**

Some of the machine learning methods such as Logistic Regression, Random Forest, Naive Bayes, and Support Vector Machines are enacted. Random Forest was found to be the most successful model and its accuracy is up to 99.78. Besides offering interpretability and a broad evaluation, confusion matrices and classification reports are offered.

5. **Integration of Artefacts**

The complete pipeline (data → preprocessing → feature extraction → model → evaluation) is written as a sequence of steps in Colab notebook steps, which can be understood as allowing the artefact to be recreated end-to-end. Use of Kaggle JSON ensures that the datasets can be scaled without necessarily requiring the datasets to be restructured in the pipeline.

To sum up, the code is not a purely experimental environment; it is a working pipeline that generates the artefact, which is an intelligent, reproducible, and automated electrocardiogram analysis system, used to produce the outcome of predicting the risk of sudden cardiac arrest.

7 Results Output

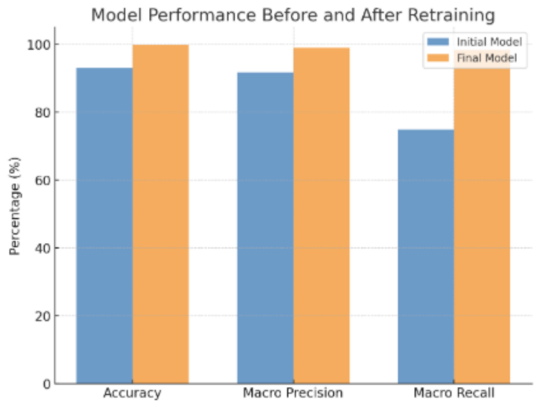

Figure 2: Best model output – Random Forest

The Random Forest classification method was first trained on this data following a thorough feature extraction process, with an initial accuracy of 93.02% attained. The macro-average recall and precision percentages were 74.94 and 91.70, respectively. After that, a bigger feature set was used for retraining, compounding the existing frequency-domain and morphological descriptors. This resulted in a considerable improvement in performance across all training sets. When tested using a separate dataset, the model produced an impressive accuracy of 99.78% along with macro precision and macro recall rates of 99.00% and 98.37%, respectively. The model's ability to distinguish between these slightly different ECG waveform patterns that characterize the classes is a result of the confusion matrix's exceptionally low misclassification.

8 Conclusion

To summarize, the current study shows that conducting of an electrocardiogram (ECG) images may be effectively digitalized, feature-engineered, and categorized using a feature-driven machine-learning pipeline. The obtained classification accuracy, interpretability of the feature correlation, as well as the robustness of the model, serve as arguments in favor of the effectiveness of the presented approach to the ECG image analysis.

References

Daydulo, Y.D., Thamineni, B.L. and Dawud, A.A. (2023) ‘Cardiac arrhythmia detection using deep learning approach and time frequency representation of ECG signals’, *BMC Medical Informatics and Decision Making*, 23(1), p. 232.

Dhyani, S., Kumar, A. and Choudhury, S. (2023) ‘Analysis of ECG-based arrhythmia detection system using machine learning’, *MethodsX*, 10, p. 102195.

Fira, M., Costin, H.-N. and Goras, L. (2024) ‘Prediction and detection of ventricular fibrillation using complex features and AI-based classification’, *Applied Sciences*, 14(7), p. 3050. https://doi.org/10.3390/app14073050

Groeneveld, R.A. and Meeden, G. (1984) ‘Measuring skewness and kurtosis’, *Journal of the Royal Statistical Society: Series D (The Statistician)*, 33(4), pp. 391–399.

Holanda, R., Monteiro, R. and Bastos-Filho, C. (2023) ‘Preprocessing selection for deep learning classification of arrhythmia using ECG time-frequency representations’, *Technologies*, 11(3), p. 68.

Dynamox ‘Vibration analysis metrics: Kurtosis and skewness’, *Dynamox*. Available at: https://dynamox.net/en/blog/vibration-analysis-metrics-kurtosis-and-skewness

Analytics Vidhya (2020) ‘Naïve Bayes algorithm’, *Medium*, 27 July. Available at: https://medium.com/analytics-vidhya/na%C3%AFve-bayes-algorithm-5bf31e9032a2