**Machine Learning for Electrode Motion artefact removal in ECG signals**

A red and white logo

Description automatically generated

Author : Ben Russell

Student ID : 40178580

Date : 02/01/2024

Supervisor : Professor Sean Mcloone

**Abstract**

This report outlines methodology followed to develop various Artificial Intelligence (AI) algorithms aimed at removing Electrode Motion (EM) noise from Electrocardiogram (ECG) signals. There are two main sections of this experiment. Firstly, the generation of a large database containing completely clean ECG signals as well as signals corrupt with a large range of EM noise was completed to train the AI algorithms. In order to generate clean signals, synthetic ECG signals were generated based on 1st order differential equations (ODE), this ensured the reference signals were completely devoid of noise and could be used as a suitable ground truth to train against. Thirty minutes of EM noise was extracted from the open source Physionet database, and an autoregressive model was used to estimate a large database of EM signals. EM signals were then linearly added to each clean ECG signal to complete the database.

The second stage is the development (training and validation) of various machine learning (ML) and deep learning (DL) algorithms. This report does not contain any technical information on this stage as it has not yet been completed, however, a full comparison of the ML/DL models will be made to standard digital filters used in industry/research.

Contents

[**1.** **Introduction** 5](#_Toc161859942)

[**1.1** **Example Digital Filters (Time/Frequency domain)** 7](#_Toc161859943)

[**1.1.1** **High-Pass Finite Impulse Response (FIR) Filter.** 7](#_Toc161859944)

[**1.1.2** **Moving Average (MA) Filter.** 7](#_Toc161859945)

[**1.1.3** **Moving Median Filter** 7](#_Toc161859946)

[**1.1.4** **Wavelet Transform Denoising** 7](#_Toc161859947)

[**1.1.5** **Empirical Mode Decomposition** 8](#_Toc161859948)

[**1.1.6** **Adaptive Filter** 9](#_Toc161859949)

[**1.2** **Example AI models** 9](#_Toc161859950)

[**1.3** **Current Databases used for Training/Validation** 10](#_Toc161859951)

[**2** **Methodology** 11](#_Toc161859952)

[**2.1** **Data Processing** 11](#_Toc161859953)

[**2.1.1** **Extraction and pre-processing of noise file.** 12](#_Toc161859954)

[**2.1.2** **Creation of synthetic clean ECG signals.** 13](#_Toc161859955)

[**2.1.3** **Creation of noisy database.** 15](#_Toc161859956)

[**2.2** **Future Work** 16](#_Toc161859957)

[**2.3** **Problems** 17](#_Toc161859958)

[**2.4** **Risks** 17](#_Toc161859959)

[**3** **Conclusion** 17](#_Toc161859960)

[**4** **References** 19](#_Toc161859961)

[**5** **Appendix** 20](#_Toc161859962)

[**Project Specification** 20](#_Toc161859963)

# **Introduction**

The Electrocardiogram (ECG) is a vital tool in modern medicine, offering a non-invasive and direct method for monitoring the electrical activity of the heart. By recording the heart’s electrical signals through electrodes placed on the skin, the ECG provides essential insights into the rhythmic patterns and conditions affecting the heart’s function. This capability makes the ECG indispensable for diagnosing various cardiac abnormalities, such as arrhythmia’s, heart disease, and myocardial infarction.

An ECG signal represents the sum of electrical potentials generated by the heart muscle during each cardiac cycle. The signal is characterized by a series of waves and complexes, most notably the p wave, QRS complex, and T wave, each corresponding to specific phases of the hearts electrical cycle. The P wave indicates atrial depolarization, the QRS complex represents the ventricular depolarization, and the T wave is associates with ventricular repolarization. Analysing these components allows healthcare professionals to assess the timing of cardiac events, the presence of abnormal rhythms, and the heath of the heart muscle.

Monitoring ECG signals is crucial for several reasons:

* **Early detection**: Continuous of periodic ECG monitoring can help detect early signs of heart disease, even before symptoms appear.
* **Diagnosis**: ECG readings are essential for diagnosing various cardiac conditions, including arrhythmia’s, ischemic heart disease, and congenital heart defects.
* **Treatment monitoring**: For patients undergoing treatment for heart conditions, ECG monitoring provides valuable feedback on the effectiveness if interventions, such as medications, pacemaker function, and recovery after cardiac procedures.
* **Prognosis**: ECG findings can inform prognosis, helping predict the likelihood of cardiac events such as sudden cardiac death or recurrence of heart attacks.

**Figure 1** below shows the typical waveform of a beat within an ECG signal.

A diagram of a graph

Description automatically generated

Figure 1 Typical ECG beat.

Despite its importance, the accuracy of ECG monitoring can be compromised by various sources of noise that can significantly affect the quality of the signal. There are 4 major sources of noise in ECG signals, these are 1) Baseline Wander (BW), 2) Powerline Interference (PL), 3) Muscle Artefact (MA) and 4) **Electrode Motion (EM)** [1]. BW and PL have a relatively unique frequency content and thus are easily removed by simple digital filters [2] (Eg notch, low pass). MA and EM noise are more challenging to remove as they can have a wide frequency content that overlaps with that of the ECG signal, specifically the PQRST complex. Adaptive filters are primarily used for these type of noise sources, however these require a reference noise signal which needs to be estimated, providing a potential source of error.

Electrode motion noise in ECG signals refers to the interference caused by the movement of electrodes attached to the skin. When electrodes move, even slightly, they can pick up additional electrical activity not related to the heart's electrical signals. This extra activity is seen as noise in the ECG signal, which can distort the true reading. It's particularly problematic during physical activity or if the electrode doesn't adhere well to the skin. This noise appears as irregular spikes or a fuzzy baseline in the ECG trace, making it challenging to accurately interpret the heart's electrical activity. Managing electrode motion noise is crucial for ensuring reliable ECG readings, especially in scenarios requiring patient movement or long-term monitoring.

EM is difficult to remove from an ECG signal for several reasons. Firstly, the signal attenuated by electrode motion can be very similar to the ECG signal itself, this similarity can make it difficult to remove with digital filters which use both time and frequency domain characteristics to separate the two signals. The variability of EM noise also adds to the difficulty to remove the noise, this comes from the wide range of movements a subject can undergo such as running, walking, jumping or any other form of activity. Each activity will return a different characteristic shape of noise. Electrode placement can also vary the shape of the noise signal. Finally, individual differences such as skin type, amount of hair and other factors influencing the electrode-skin contact will affect the amount of EM noise added during movement, it is important that algorithms can deal with this problem effectively.

## **Example Digital Filters (Time/Frequency domain)**

As discussed in the previous section, there has been an ample amount of research into the design of digital filters to try and mitigate the impact of noise on the ECG signal. Digital filters are broadly divided into two classes: finite impulse response (FIR) and infinite impulse response (IIR). In biomedical signal processing, FIR filters are generally used for their linear phase advantages. EM presents a challenge to all current filters, but can be reduced by High Pass FIR, moving average, moving median, wavelet transform filters, empirical mode decomposition and adaptive filters. A high-level description of these filters is presented below.

### **High-Pass Finite Impulse Response (FIR) Filter.**

Described by the following discrete difference equation:

(1)

Where is the coefficient of the filter, M-1 is the order of the filter, M is the length of the filter, x(n) is the input signal and y(n) is the filtered signal.

### **Moving Average (MA) Filter.**

Described by the following discrete difference equation:

(3)

(4)

Where denotes the noisy ECG signal, denotes the estimated noise, is the ECG signal after denoising, M is the filter length.

### **Moving Median Filter**

Same as MA but uses median instead of average.

### **Wavelet Transform Denoising**

Wavelet Transform denoising uses the discrete wavelet transform (DWT) to decompose the ECG into several time-domain signals at different frequency bands, then sets the approximation coefficients at the lowest frequency band to zero and reconstructs the ECG signal by synthesizing the modified coefficients. Figure 2 demonstrates this where the signal x(n) is decomposed into detail coefficients d(n) and approximation coefficients a(n). The approximations coefficients are related to the low-frequency part of the signal while the detail coefficients are related to the high-frequency components.

A diagram of a flowchart

Description automatically generated

Figure 2 Demonstration of 3-level discrete wavelet transform decomposition.

### **Empirical Mode Decomposition**

Described by equation (5) :

(5)

Where represents the intrinsic mode function and represents the residual signal.

Empirical Mode Decomposition (EMD) decomposes a signal into a number of intrinsic mode functions (IMF) and a residual signal using the shifting process. Some of the IMFs contain useful signal information, while other contain signal and noise and so it can be used for denoising.

### **Adaptive Filter**

**A diagram of a computer algorithm

Description automatically generated**Adaptive filters are time-variant filters which is self-designed and has the ability to adjust its’ parameters automatically according to an optimisation algorithm. It has the capability of adaptively tracking the signal under non-stationary conditions. Figure 3 presents a high level block diagram of an adaptive filter.

Figure 3 Block diagram of adaptive filter.

Adaptive filters require two sets of input signals: The primary input and the reference input, where the primary input d(n) contains the noise corrupt signal. The reference signal S(n) is correlated with the noise, but uncorrelated with the signal. The reference signal is fed into a digital filer to produce an output y(n), which is as close as possible to the noise N(n). The coefficients of the filter are continuously updated according to the chosen adaptive algorithm. Three common adaptive algorithms are the Least Mean Square (LMS), Normalised Least Mean Square (NLMS), and Recursive Least Square (RLS).

## **Example AI models**

The need for intelligent models to try and reduce EM noise in ECG signals arose due to the limitation of using traditional time/frequency domain on non-stationary and non-linear noise sources (Muscle Artefact and Electrode Motion). This is a relatively new area of research, however, there have been attempts by researchers to develop models that aim to learn the characteristic shape of the specific type of noise. Namely, Brophy et all [5] proposed a deep learning framework where they employ a custom loss function to train a Convolutional Neural Network (CNN). Results look promising; however, performance was not yet good enough to be used over standard digital filters. Corneliu et all [6] present a review on DL models for denoising ECG signals. Two deep learning models are investigated; 1) A CNN with six convolutional layers, with subsequent pooling and a fully connected layer for regression and 2) A Long-Short term memory (LSTM) model, consisting of 2 LSTM layers. Results suggest that CNN’s are superior to LSTM models in denoising ECG signals. The objective of this report is to enhance performance reported by Brophy by 1) Using a larger, and more diverse training dataset, and 2) Employing a new loss function called Wavelet Energy Based Diagnostic Distortion [7] which should capture the diagnostic information contained in the ECG waveform.

## **Current Databases used for Training/Validation**

The MIT-BIH Arrhythmia database has been used alongside the Noise Stress Test Database (NSTDB) to create a database of noisy ECG signals. The MIT-BIH files are assumed as the clean records, however in reality, these are not clean signals. This report will aim to improve on this area by generating synthetic ECG signals that are completely devoid of noise, this ensures that the model is being trained against appropriate ground truths.

The performance of the developed algorithms will be compared to that of standard digital filters mentioned in section **1.1**. The performance will be established using a variety of objective metrics, specifically SNR improvement, Cross-Correlation and WEDD. Visual inspection of the denoised ECG signals will also be looked at.

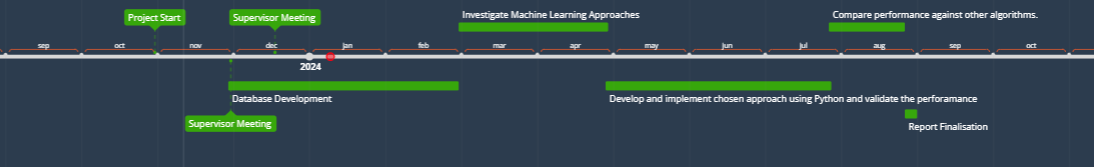
Figure 4 shows the full project planning timeline, as well as the current position.

Figure 4 Project timeline and current position marked on it.

**Current Position**

All work on this project is being tracked and version controlled using Github. A repository has been created [8] that stores all code and documentation for the project, this enables effective tracking of databases, models and reports. Sourcetree has been used as a graphical user interface (GUI) for Git which enables both developers and reviewers to easily look at changes throughout the project life cycle. All commitments to code and reports have been described in the corresponding repository commit, this is shown in figure 5 below:

A screenshot of a computer

Description automatically generated

Figure 5 Snipped of SourceTree history interface. Commits along with the date can be seen clearly.

As of 09/01/24, The project contains a mixture of MATLAB and Python code. MATLAB has been used to generate the reference database on which the model will be trained, this is due to MATLAB’s enhanced ability for signal processing problems where noise is being modelled and added to clean ECG signals. MATLAB’s ability to work with matrices makes this relatively simple. Python is used to develop machine learning models, using the reference database. To date, only a reference deep learning model has been added to the repository. Python has numerous frameworks specifically designed for implementing AI models (Pytorch, Tensorflow) and so has been selected as an appropriate language to use.

# **Methodology**

This section aims to dive further into the data processing techniques used in the dataset development, since this stage is almost complete.

## **Data Processing**

The following flow diagram presents a high level overview of the data processing steps followed to generate the noisy database used for training various intelligent algorithms:

**createSyntheticCleanEcgSignals.m**

**ECG signals output as .mat files in a unique directory**

**generatingNoisyEcgDatabase.m**

**Single 30 minute noise record output as .mat file in unique directory**

**extractAndPreProcessNoiseSignal.m**

**createTrainingDataset.m**

**Figure 6 Flow Diagram over dataset creation.**

### **Extraction and pre-processing of noise file.**

A blue line graph with white text

Description automatically generated with medium confidenceElectrode motion noise was downloaded from Physionets Noise Stress Test database (NSTDB) [3], along with the sampling frequency of the noise. The figure below shows the raw noise signal itself:

**Figure 7 Electrode Motion noise obtained from Physionet. The sampling frequency is 360Hz,**

A red line graph on a white background

Description automatically generatedThe ECG signal was then resampled to 500Hz using MATLAB’s native resample.m function, this was done to ensure all processing in the database creation is complete at 500Hz. This noise signal shown in figure 4 was then passed through a 20th order Chebyshev Type II bandpass filter to remove the baseline noise that can be seen, the cut-off frequencies were set at 1Hz and 50Hz such that only EM noise remained in the signal. The processed signal can be seen below:

Figure 8 Post-processed electrode motion noise data.

### **Creation of synthetic clean ECG signals.**

To create a dataset of clean ECG signals, Physionets ECGSYN tool was utilised. This tool allows a user to generate a clean ECG signal where they can choose the morphology of the beats. In this experiment, and as with most Intelligent models, it is important that the algorithm is trained on as large a database as possible, with as many different characteristic shapes of the time series data.

ECGSYN allows for the alteration of the following features:

1. Angles of Extrema – Deflection angle of each waveform.
2. Z Position of Extrema – Amplitude of each waveform.
3. Gaussian Width – Width of each waveform.

This experiment defined the maximum/minimum angle of extrema possible as:

This experiment defined the maximum/minimum Z of extrema possible as:

This experiment defined the maximum/minimum width possible as:

Where the arrays refer to the parameter setting for a specific feature ([P, Q, R, S, T]).

Latin Hypercube sampling [9] was employed to generate 10,000 random samples between the defined limits. This ensured that each signal generated had different morphologies, and that each morphology generated was realistic. By realistic we mean the waveforms are technically possible to be generated by a human heart.

It was also necessary to generate signals with different Heart Rates (HR), the available HR’s were set as:

We generate 10,000 signals per HR and thus, the final dataset will have 60,000 ECG signals.

**Table 2 Algorithm design of clean ECG signal database.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm 2: Algorithm to construct clean ECG signals.** | | | |
|  | ***Input***:  HR\_TO\_GENERATE  MIN\_ANGLES\_OF\_EXTREMA  MAX\_ANGLES\_OF\_EXTREMA  MIN\_Z\_POSITION\_OF\_EXTRAMA  MAX\_Z\_POSITION\_OF\_EXTRAMA  MIN\_GAUSSIAN\_WIDTH  MAX\_GAUSSIAN\_WIDTH | | |
|  | Output: A database of clean ECG signals with different morphologies and Heart Rates. | | |
| **1** | Sampling 🡨 Generate 10,000 random sample parameters of our inputs. | | |
| **2** | For (iHeartRate = 1 : numberOfHeartRates) // Loop through all heart rates defined. | | |
| **3** |  | Define the HR to generate 10,000 signals for. | |
| **4** |  | for (iSignal = 1 : numberOfSignalsPerHR) // Loop through each of the 10000 signals | |
| **5** |  |  | **Define parameter settings:** *Waveform angle, waveform amplitude, width size.* |
| **6** |  |  | ***Try:***  *Call ECGSYN function with specific parameter settings.*  ***Catch:***  ***Continue***  ***end*** |
| **7** |  |  | ***Save :*** *Save the signal with HR and signal number in filename.* |
| **8** | ***Save :*** *Save the parameter setting matrix.* | | |

### **Creation of noisy database.**

At this stage, there exists a 30-minute EM noise file contained in a directory, and 60, 000 ECG signals contained in another directory. The aim of this processing step was to combine the noise and ECG signals such that a noisy signal was produced for each ECG shape. This experiment generated noise at Signal to Noise Ratios of 0, 6, 12, 18 and 24 decibels. 30 second ECG strips were generated, and 10 noisy signals were generated per clean ECG, details of this are provided below.

The noise signal is first read into MATLAB, we then loop through 30 second segments of the 30-minute noise file and model further shapes of the noise, this is done to extend the noise shapes that may not be contained in the 30 minute strip. This is done using an auto-regressive model [10] which is used to predict further noise shapes.

We then iterate through each clean ECG signal generated and read in the corresponding R-Peak positions. (These are returned from ECGSYN). The R-peak locations are used calculate the peak-to-peak amplitude and power, which in turn allow the specified SNR levels to be determined.

The overall algorithm can be seen below:

**Table 3 Algorithm design of noisy database creation.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm 3 : Algorithm to construct Noisy ECG signals.** | | | |
|  | ***Input***: Path to noise signal  Path to clean ECG signals  Sampling frequency  ECG Length  Number of noise sections  SNR Levels  Number of generated noise signals. | | |
|  | Output: A database of noisy ECG signals with different morphologies and Heart Rates. | | |
| **1** | Load Noise Data 🡨 Read in 30 minute noise signal | | |
| **2** | Segment the noise signal into 30 second strips. | | |
| **3** |  | for (iSegment = 1 : Number of noise sections)  Predict a new 30 second noise strip using auto-regressive modelling. | |
| **4** |  |  | ***Save :*** *Save each noise signal to a table on MATLAB.* |
| **5** | **for (iCleanEcgSignal = 1 : numberOfEcgSignal)** | | |
| **6** | **Load ECG Data 🡨** Read in 30 second record. | | |
| **7** | **Load QRS Locations 🡨** Load in the QRS locations. | | |
| **8** | **Calculate** the peak to peak amplitude of QRS peaks. | | |
| **9** | **Convert** peak to peak amplitude to power | | |
| **10** | **Scale** the noise signal to each required SNR level. | | |
| **11** | **Add** the noise signal to the clean ECG record. | | |
| **12** | **Save** : Save the table which contains all SNR levels and noise corrupt  Signals for one clean ECG record. | | |
| **13** | **end** | | |
|  |  | | |

## **Future Work**

The next stage of this project is to finalise the training dataset, and verify that all signals look like realistic ECG waveforms, this will be done manually by visually looking at a subset of the signals. From there, the development stage can begin. Machine learning algorithms will first be implemented; however this will be to demonstrate how these types of algorithms are not suited towards the objective of this paper. It is expected that deep learning will provide a vastly superior method to remove these complex, non-stationary time signals, and so most of the models will be variants of deep learning algorithms (RNN, CNN etc). The training of these algorithms will need to be complete with the use of a Graphical Processing Unit (GPU) due to the large amount of data, this will be done using QUB’S Kelvin system.

The development of the algorithms will be complete using the Pytorch Framework [11s] which provides useful features such as tensors to store temporary data. All models will be uploaded to the GitHub repository listed previously. After models are trained, the performance will be demonstrated on 2 new datasets; 1) New simulated data will be generated producing the clean and noisy counterpart signals, and 2) MIT-BIH Arrhythmia database from Physionet. The objective is to demonstrate that the models generalise to both unseen synthetic and real-life data. Performance metrics such as WEDD, MSE and cross-correlation will be used to assess the distortion between the processed and clean ECG signals.

## **Problems**

There were several problems encountered in the database development. The parameter setting stage when generating clean signals has been difficult as they have been set randomly based on a reasonable estimation, however in practice these may not be realistic. Further research and verification of generated signals will be required to ensure the algorithms are not trained on unrealistic ECG waveforms. Furthermore, the amount of data has presented a new challenge. Due to the amount of ECG signals being generated, it is difficult to store these on local machines and thus a larger computer with GPU access needs to be used. QUB’s Kelvin system will be utilised to solve this problem.

## **Risks**

There are no major risks associated with the project, and the project is on track as it currently stands. However, there may be problems arising in the data creation that could results in no convergence of any of the models, which obviously would prevent the project completion. Incorrect or corrupt training data would likely push the project timeline back as these would need to be investigated. To mitigate this, a verification stage will happen after the database has been created to check the data looks correct. There is also the risk of not being able to access the Kelvin system to train the model, this would mean the models would have to be trained on a subset of the data, putting the accuracy of the models at risk.

Less likely risks such as machines being used breaking, or accidently deleting data/files will be mitigated by keeping all files/data on an online repository. This means they can be accessed from different machines (cloud storage). A spare laptop is kept in case of failure.

# **Conclusion**

This report presents the strategy being followed to develop an Intelligent model that can effectively remove electrode motion noise from an ECG signal while maintaining the underlying clean ECG signal. Various algorithms designed to create clean ECG signals and corrupt them with various amounts and characteristic shapes of electrode motion noise are presented and some results are shown. No learning models have been developed to date; however, the project timeline is presented that clearly shows when development of the algorithms will occur.

# **References**

[1] Kumar P, Sharma VK. Detection and classification of ECG noises using decomposition on mixed codebook for quality analysis. Healthc Technol Lett. 2020 Feb 18;7(1):18-24. doi: 10.1049/htl.2019.0096. PMID: 32190336; PMCID: PMC7067057.

[2] Rahul Kher (2019) Signal Processing Techniques for Removing Noise from ECG Signals. J Biomed Eng 1: 1-9.

[3] Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220.

[4] McSharry PE, Clifford GD, Tarassenko L, Smith L. A dynamical model for generating synthetic electrocardiogram signals. IEEE Transactions on Biomedical Engineering 50(3): 289-294; March 2003.

[5] E. Brophy, B. Hennelly, M. De Vos, G. Boylan and T. Ward, "Improved Electrode Motion Artefact Denoising in ECG Using Convolutional Neural Networks and a Custom Loss Function," in *IEEE Access*, vol. 10, pp. 54891-54898, 2022, doi: 10.1109/ACCESS.2022.3176971. keywords: {Electrocardiography;Electrodes;Noise reduction;Signal to noise ratio;Convolutional neural networks;Recording;Signal denoising;Convolutional neural network;custom loss function;electrocardiography;signal denoising}

[6]

[7] M. Sabarimalai Manikandan, S. Dandapat, Wavelet energy based diagnostic distortion measure for ECG, Biomedical Signal Processing and Control, Volume 2, Issue 2.

[8] [MSc-Project/Electrode Motion Denoising at main · ben120-web/MSc-Project (github.com)](https://github.com/ben120-web/MSc-Project/tree/main/Electrode%20Motion%20Denoising)

[9] <https://en.wikipedia.org/wiki/Latin_hypercube_sampling>

[10] <https://www.mathworks.com/help/signal/ref/arburg.html>

[11] https://pytorch.org

# **Appendix**

# **Project Specification**

Table 1 indicates the relevant sub-field that are employed in this project.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Control | X | Embedded Systems |  | High Frequency Electronics |  | Microelectronics |
|  | Electric Power | X | Software | X | Connected Health |  | MEMS |
|  | Cyber-Security |  | Wireless Communications | X | Signal/Image Processing | X | Intelligent Systems |
|  | Digital Design |  | Sensor Networks | X | Data Analytics | X | Electronics |

Table 1 Shows the sub-fields that are relevant to this MSc project

The electrocardiogram (ECG) is a non-invasive method to measure the electrical activity of the heart and can be used to diagnose heart disease. According to the World Health Organisation (WHO), chronic heart disease was the number one cause of death from 2000 – 2019. Long term ECG monitoring is currently the gold standard for diagnosing cardiovascular diseases (CVDs), however obtaining reliable long-term measurements of the ECG signal is challenging because patients are required to collect their ECG signal remotely on a wearable device. Wearable devices are inherently contaminated with noise which can supress the essential pathological biomarkers and, in some cases, render the ECG completely unusable. ECG signals can be contaminated by many types of noise including: 1) Baseline Wander, 2) Powerline Interference, 3) Electromyographic and 4) Electrode Motion artefacts. Most of these noise sources can be reduced through the use of time and frequency domain digital filters. However, the frequency spectrum of electrode motion noise overlaps with the frequency spectrum of a typical ECG signal making it very difficult to remove in the time and frequency domain. The objective of this project is to explore if AI/ machine learning can be used to learn the characteristics of, and correct for, electrode motion induced noise on ECG signals.

**Objectives**

1. Conduct a literature review on ECG motion artefact removal algorithms to identify the different approaches that exist, and the challenges involved in developing effective methods and assessing their performance.
2. Using available online sources (e.g. Physionet1) compile an ECG dataset which can be used to investigate and assess ECG motion artifact removal algorithms.
3. Develop a model for generating synthetic motion artefacts in clean ECG signals and use it to create a reference dataset for ground truth comparisons.
4. Investigate machine learning approaches to reducing electrode motion noise on ECG signals.
5. Develop and implement a candidate approach using Python or Matlab and validate its performance on the datasets from (2) and (3).
6. Compare the performance of the developed algorithm against alternative baseline algorithms from the literature.

**MEng Extension**

1. Explore advanced deep learning concepts (e.g., transfer learning, data augmentation) to enhance the performance of models and/or develop and evaluate alternative machine approaches for motion artefact removal.
2. Provide a rigorous assessment of all approaches developed with regard to real-time/embedded system implementation constraints.

**Learning Outcomes**

At the end of the project the student will be able to demonstrate:

1. A good understanding of ECG denoising algorithms.
2. A working knowledge of machine learning/ deep learning models and associated development tools
3. Enhanced programming skills in Python or Matlab, particularly with regard to algorithm development and signal processing.