HIGH SCHOOL EDITION
Journal of Student Research
Volume 12 Issue 3 (2023)

Utilising Machine Learning to Predict Myocardial Infarction by Electrocardiogram Derived Respiration

Evelyn Fung[1] and Shadi Ghiasi[#]

[1]The Cambridge Center for International Research
[#]Advisor

ABSTRACT

Myocardial Infarction (MI) is one of the leading causes of death. Electrocardiogram (ECG) is a non-invasive tool that is commonly used as a diagnostic tool to assess cardiac conditions. A dataset consisting ECG signals of healthy individuals and MI patients was subjected to pre-processing techniques like normalization and application of a bandpass filter. R-R peak intervals from the pre-processed ECG signals are extracted to generate the respiratory signal. The features extracted from the respiratory signal are used to predict MI. The objective of this study is to evaluate the potential of ECG derived respiratory signal (EDR) in predicting MI by utilizing machine learning techniques like random forest, linear regression, Convolutional Neural Network(CNN), Multilayer perceptron(MLP). The results of the study will examine the feasibility of using EDR in predicting MI and provide insight into the most effective machine learning technique for this application. This study will contribute to the development of new and efficient prediction methods for MI patients.

Introduction

Real-time biomedical signals is an emerging field with great potential in healthcare [1]. The current signal processing tools or programs are more suited for engineers working in biomedical applications because they can use them readily given their position [2]. These biological signals has largely concentrated on the diagnosis and interpretation of data on a person's health status, which can be used to detect diseases such that patients can receive treatment before the situation worsens.

Myocardial Infarction (MI) occurs when a lack of oxygen in heart tissue causes eventual death of that tissue if the supply does not get restored within a short time [3]. If the blood flow to a part of the heart decreases, it can lead to ischemia as less oxygen supplied for respiration, resulting in tissue death.

The early detection and prediction of MI is one of the major difficulties in the healthcare industry. Electrocardiogram (ECG) has been widely used for predicting and diagnosing MI. Recently, the development of machine learning have played a significant role in diagnosing various diseases. One example is the use of ECG signals to extract features to classify heart conditions [4]–[6]. An ECG is a quasi-periodical, rhythmical signal produced by the functioning of the heart, acting as the source of bioelectrical events. This signal can be represented as an electric dipole and it can be measured by placing a detector on the surface of the body. The normal ECG consists of P wave, QRS complex, and T wave. The QRS complex can be broken down into three components - Q wave, R wave, and S wave. The P wave is produced by the electrical signals generated during atrial depolarization before contraction. The QRS complex is created by the electrical signals during ventricular depolarization before contraction. The T wave is due to the electrical signals produced during ventricular recovery from depolarization, also known as a repolarization wave. The ECG reflects both depolarization and repolarization. The lead V4 is used for single-lead analysis of ECG in this paper.

MI results in an elevation on the ST segment in the front leads (V3 and V4) at the J point, and sometimes in the septal or lateral leads depending on the severity of the MI. This ST segment elevation is concave downward and

often overshadows the T wave [7]. The fragmented QRS complexes (fQRSs) is characterised by the abnormal shape of the QRS complex and fragmentation or notching of the QRS wave. The fQRS complex may become wider, indicating delayed or abnormal ventricular depolarization. The fQRS complex may also exhibit deeper notching or fragmentation, indicating changes in the conduction system of the heart. These are markers of altered ventricular depolarization owing to an incidence of MI. [8]

Many researchers have attempted to extract a patient's respiratory signal using signal processing of the ECG. This respiratory signal, known as the ECG-derived respiration (EDR) signal, has been found to have various clinical implications [9]. The aim of ECG-based respiration measurement is to extract information related to respiration from an ECG signal without the need for a hardware. The respiration rate signal derived from single-lead ECGs is a crucial for the analysis of Heart Rate Variability (HRV). The EDR technique is based on the slight variations in ECG morphology that occur during the respiratory cycle, caused by changes in the position of the heart relative to the electrodes and alterations in lung volume [10].

Currently, there are studies that predicts MI using machine learning with a high accuracy, which requires the long-term monitoring of subject's ECG. However, little attention has been paid to the prediction of MI using EDR. In this study, a 20s window from a 1 minute ECG signal is converted to EDR for the prediction of MI using features extracted from EDR. Short-term monitoring of ECG can be more effective in predicting MI than long-term monitoring, as it captures the significant changes in ECG that occur during the acute phase of an MI. This makes it more suitable for implementation into an instantaneous alert monitoring system, allowing for early detection in case of a potential MI.

Materials and methods

Dataset Description

The study utilized the PTB Database ECG Database, which is part of the commonly used MIT-Physionet database for ECG signal research, as its data source. The database consists of 549 records from 290 subjects, 148 of which are annotated for MI. Each record includes 15 simultaneously measured signals: the conventional 12 leads (i, ii, iii, avr, avl, avf, v1, v2, v3, v4, v5, v6) together with the 3 Frank lead ECGs (vx, vy, vz). The digitized ECG signals are recorded at 1000 samples per second and are represented in 16 bits per sample over a range of ± 16.384 mV. For training purposes, the study used a subset of the dataset, which consisted of 1 minute of 10 ECG recordings of healthy and of 10 ECG recordings of patients with myocardial infarction at 1000 Hz. Of these recordings, the data is apportioned into training and test sets, with a 80-20 split. To provide a fair comparison, the ECG signals are taken from 5 female and 5 male for healthy and MI ECG.

Data pipeline

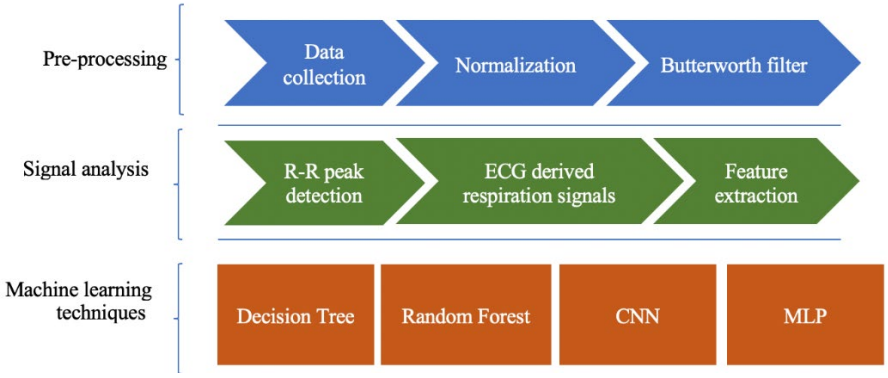

Fig. 1: Pipeline for the study for MI detection

The data is segmented into 1 minute ECG signals after obtaining data from PTB Database. The signal is then filtered with pre-processing techniques as shown in Sect IIC. RR peaks are extracted to calculate the R-R interval and generate an EDR. Features are extracted from EDR and inputted into the machine learning for prediction of MI.

Data Pre-processing

The ECG and respiratory signals can be influenced by various types of unwanted artifacts, like baseline wander, power line noise, and interference from muscle activity. Baseline wander is a low-frequency noise due to patient's movement and has a frequency range from 0.5 to 0.6 Hz [11]. Interference from electrical appliances can cause a 50 Hz noise in the signal. These noise can result in incorrect interpretations of physiological signals.

Several pre-processing techniques were applied to the ECG signals in order to obtain the optimal data for analysis. The first step was to remove baseline wander which was done through normalization using mean and standard deviation using the following equation.

$$z = \frac{x - \mu}{\sigma}$$

where μ is the mean and σ is the standard deviation

The next step was to design a second order Butterworth filter with a range of 0.01 to 50 Hz, which is the standard cut-off frequency for ECG(ask for source), and a sample frequency of 1000 Hz. The filter was implemented using the Scipy library in Python. Lastly, the signals were normalized again using the formula above. The aim of these preprocessing techniques is to remove any unwanted noise from ECG signals, resulting in a clean signal for analysis.

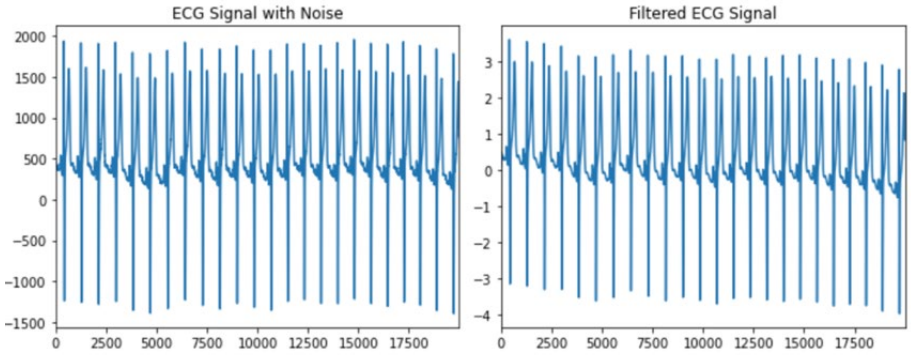

Fig. 2: Raw ECG vs Filtered ECG

R-R peak detection

R-R peak detection is performed on an ECG signal. The ECG signal is used to compute heart rate (HR) by determining the R-R intervals, which is the time between consecutive R peaks in an ECG signal. The first 20 seconds of the filtered signal are used for peak detection. The find peak function in scipy.signals library is used to detect the peaks in the signal. A peak or local maximum is defined as any sample whose two direct neighbours have a smaller amplitude [12]. The required of peak height is set at 0.5 and required minimal horizontal distance between peaks is 500 samples.

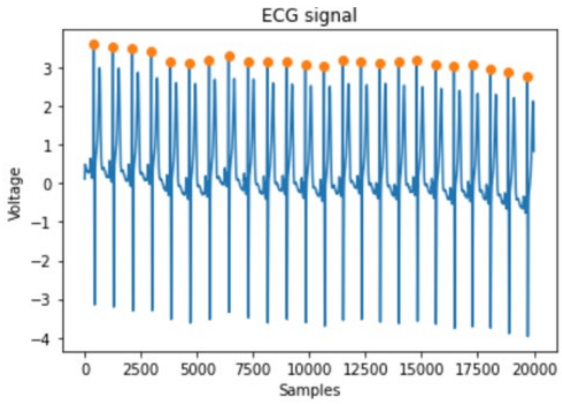

Fig. 3: ECG signal with R peak detection

Estimation of ECG Derived Respiration EDR signals

R wave duration (RWD) is calculated by measuring the time between two inflection points next to each R wave peak. A EDR signal plot was generated by plotting the RWD values against time. [13] The respiration signal usually falls within a frequency band of 0.1 to 0.7 Hz. As the respiration cycle is a slow process and requires 6 to 10 heartbeats to generate one respiration cycle, linear interpolation can be used to increase the rate of signal reconstruction. These intervals are interpolated using a cubic spline interpolation. In this case, the sampling frequency of the R-R interval signal is increased to 4, which means that the function will interpolate the RRI signal at four times the original sampling frequency. This results in a more finely sampled RRI signal, which can be useful for certain applications such as heart rate variability analysis. To get a waveform similar to the original respiration signal, cubic spline interpolation is used. Suppose for a data set of original signals, $f_i = f(x_i)$ [where $i = [0,1,2,...n-1]$, the output of the cubic-spline interpolation at a given interval (x_i, x_{i+1}), f can be written as the following equation as shown below. [14]

$$f = pf_i + qf_{i+1} + rf_i'' + sf_{i+1}''$$

Where,

$$p \equiv \frac{x_{i+1} - x}{x_{i+1} - x_i}, \qquad q \equiv \frac{x - x_i}{x_{i+1} - x_i}$$

$$r \equiv \frac{1}{6}(p^3 - p)(x_{i+1} - x_i)^2, s \equiv \; r \equiv \frac{1}{6}(q^3 - q)(x_{i+1} - x_i)^2$$

Note: p and q are linearly dependent on x and r. s have cubic-x dependence.

For first derivative,

$$\frac{df}{dx} = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} - \frac{3p^2 - 1}{6}(x_{i+1} - x_i)\, f_i'' + \frac{3q^2 - 1}{6}(x_{i+1} - x_i)\, f_{i+1}''$$

For second derivative,

$$\frac{d^2 f}{dx^2} = pf_i'' + qf_{i+1}''$$

The interpolated R-R intervals are then used to compute the heart rate in beats per minute(bpm). The HR signal is detrended and normalized to remove any variation, and respiratory peaks are then detected from the resulting signal.

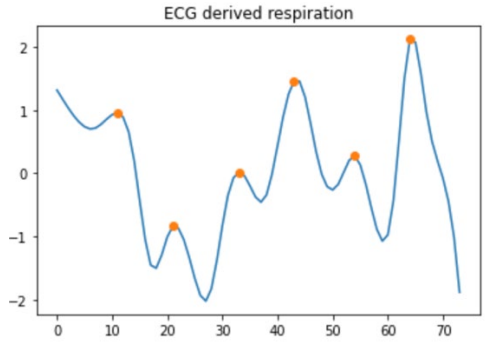

Fig. 4: ECG derived respiration(EDR)

Power spectral analysis

Heart rate variability power spectral analysis is widely used in the medical field to assess the neuro-cardiac control mechanism. The Fast Fourier Transform (FFT) is commonly used in power spectrum analysis. It transforms a time-domain signal into its frequency-domain representation. The FFT algorithm breaks down the Discrete Fourier Transform (DFT) computation into smaller sub-problems. The advantage of this method include easier use in nonstationarity tests and a reduction in computations and core storage requirements [15]. The FFT algorithm is based on the equation for DFT.

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-i2\pi nk/N}$$

where $x[n]$ is the time-domain signal, $X[k]$ is the frequency domain representation of the signal, N is the total number of samples, k is the frequency bin, and i is the imaginary unit.

This equation computes the contribution of each time-domain sample to each frequency bin of the resulting frequency domain representation.

There has been research which studied EDR using frequency analysis [16]. The most distinct peak reflects changes in beat to beat interval that oscillates at the same frequency as respiration. The power in the high frequency band (0.4-1 Hz) has shown potential in detecting coronary artery disease [17]

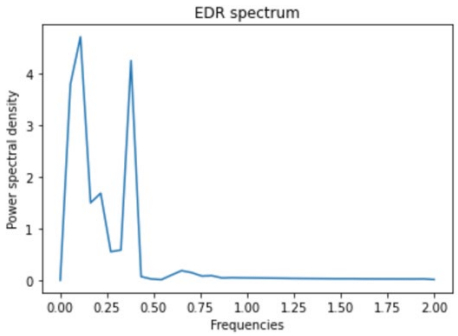

Fig 5: EDR spectrogram for MI patients

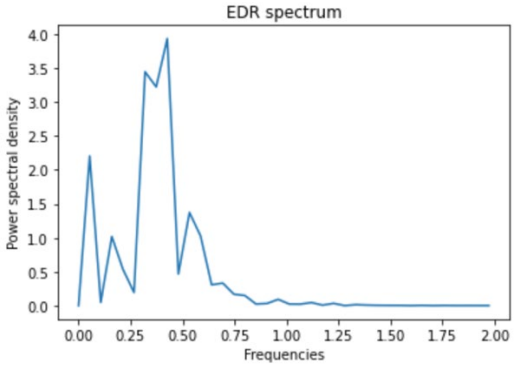

Fig 6: EDR spectrogram for healthy patients

Feature extraction

The figure below shows the feature extracted from the EDR, which is inputted into the machine learning algorithms.

Table 1: List of extracted EDR features for this study

Parameter	Unit	Description
HR(mean)	BPM	Mean of heart rate
RR(mean)	Hz	Mean of respiratory rate
RP(mean)	s	Mean of respiratory period
IBI(min)	s	Minimum of inter breath peak time interval
BM(mean)	s	Mean of the interval between local minimum of each breath
FFT(peak)	Hz	Maximum frequency of FFT spectrum in periodogram

Machine learning

To classify and predict patients with and without MI using machine learning, Python is used and various libraries such as sklearn, seaborn, tensorflow.keras, and matplotlib are also imported. In order to identify the most effective machine learning models, different models were tested on our dataset to predict MI. This study uses 2 classification machine learning algorithms (Decision tree, Random forest) and 2 deep learning algorithms (Convolutional Neural Network, Multilayer perceptron). Cross-validation, accuracy and loss curves for deep learning models, and confusion matrices were applied to evaluate the prediction accuracy. The dataset was randomly split into two sets, with 70 percent for training and 30 percent for testing. Below are the machine learning classifiers used, along with their descriptions.

Decision Trees

Decision Trees is a powerful machine learning algorithm that utilizes a tree-based structure to mimic human's thinking process by modelling decisions based on the input data. The tree-based structure divides the data into smaller and smaller subsets based on the feature values. Each internal node in the tree represents a test on a feature, and each leaf

node represents a prediction or decision. The main idea behind Decision Trees is to use an algorithm to repeatedly split the data into subsets based on the feature values, until a stopping criterion is reached, such as the minimum number of samples per leaf node, or a maximum tree depth. The criterion used for splitting is typically the information gain or Gini impurity, which measures how well the split separates the data into the target classes. The criterion is set to ”entropy” [18], which uses the entropy of the data to calculate the impurity of a split while the maximum depth of the tree is set to 3, which means the tree will only split 3 times.

Random Forest

A variation on decision trees called a random forest creates many trees and combines them to get a final prediction. The predictions of each tree in Random Forest are averaged to get the final prediction. Each tree in Random Forest is developed on a randomly selected portion of the data and features. By integrating the predictions of various trees, each of which has limited access to the data and features, Random Forest aims to reduce overfitting. Furthermore, Random Forest is resilient to the presence of noisy or unimportant features and can handle feature interactions and non-linear correlations. The number of trees, the size of the random subsets, the maximum tree depth, and the splitting criterion are all variable factors in Random Forest. The parameters set for this method are identical to those used in decision trees.

Convolutional Neural Network (CNN)

CNNs are a class of deep neural networks designed to handle image and video data. The main building blocks of CNNs are convolutional layers, which perform a convolution operation between the input image and a set of filters, and pooling layers, which downsample the data to reduce its dimensionality. The filters in the convolutional layer are learned during training, and they are responsible for detecting different patterns and features in the image. The pooling layer reduces the spatial dimensions of the data and helps to reduce the computational complexity. The final layers of the CNN are fully connected layers, which take the features detected by the convolutional and pooling layers, and produce the final prediction. The parameters that can be tuned in CNNs include the number of filters, the size of the filters, the size of the pooling, and the activation function used in the fully connected layers. A dense layer with 12 neurons and a rectified linear unit (ReLU) activation function is created. It works by thresholding valuesat 0, i.e.$f(x) = max(0,x)$. Simply put, it outputs 0 when $x < 0$, and conversely, it outputs a linear function when $x \geq 0$. 6

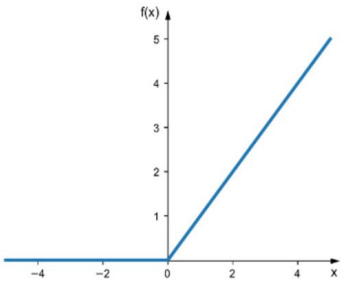

Fig. 7: Rectified Linear Unit (ReLU) activation function[19]

After applying this function, another layer with a single neuron and a sigmoid activation function is applied. The loss function is used to evaluate the model’s performance. In this case, the loss function is set for binary classification problems. The optimizer is set to ”adam”, which is a popular optimization algorithm for deep learning models. The Adam optimization algorithm is used to minimize the loss function and adjust the weights of the network. It is used for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. [20]

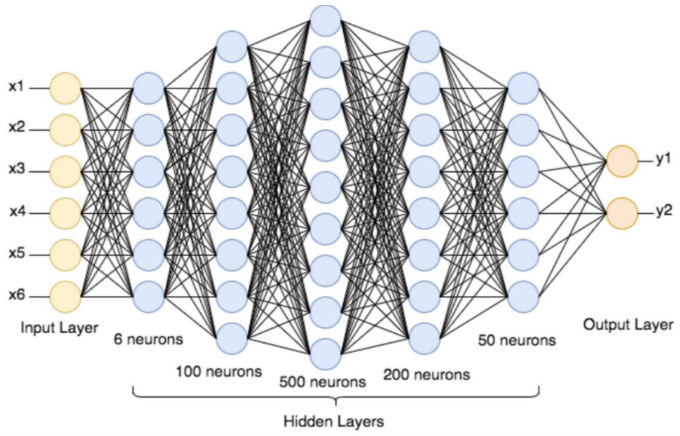

Fig. 8: Deep Neural Network architecture[21]

Multi-layer Perceptron

An artificial neural network called a multi-layer perceptron (MLP) is used in supervised learning, including binary and multi-class categorization. This algorithm employs numerous layers of artificial neurons—interconnected nodes—to process the incoming data. Each neuron in one layer takes inputs from several neurons in the previous layer, computes using these inputs, and then produces the result to the next layer. There is at least three layers, including an input layer, one or more hidden layer(s), and an output layer, make up each MLP [22]. The number of hidden layers, the number of neurons in each hidden layer, the activation function used in the hidden layers, and the loss function used to train the network are additional parameters of the MLP.

There are 12 neurons in one hidden layer in this MLP model. The first and second layers of the MLP model each have 64 and 32 neurons, respectively. The input layer, which has 9 neurons (the number of features in the input data), is connected to each of the neurons. The activation function used in the hidden layer is the rectified linear unit (ReLU), which replaces negative inputs with zeros. The output layer has a single neuron and uses the sigmoid activation function, which maps the input values to a range between 0 and 1, suitable for binary classification.

The MLP model is then compiled using the binary cross-entropy loss function, which measures the difference between the true target values and the predicted values. The Adam optimization algorithm is applied here. Finally, the model is trained on the training data for 10 epochs, where an epoch is a complete iteration through the entire training dataset. The accuracy of the model is monitored on the test data during training.

Results

Decision Tree and Random Forest results

When the values in a set are close to together, the set is considered as precise. If the average is close to the true value of the quantity being measured, the set is considered accurate. To determine both precision and accuracy, it is necessary to have a set of data points from repeated measurements of the same quantity.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Where *TP* is the number of true positives, *TN* is the number of true negatives, *FP* is the number of false positives and *FN* is the number of false negatives.

For both Decision Tree and Random Forest, a confusion matrix of the following was obtained. 0 represents the patient is predicted to have no MI while 1 represents the patient is highly likely to have MI. Out of 6 of the testing data, 1 of them was wrongly predicted.

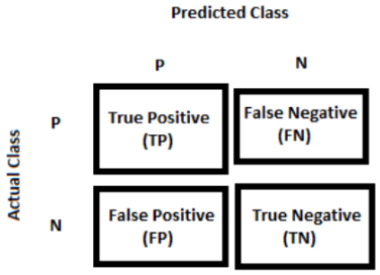

Fig. 9: Confusion Matrix sample [23]

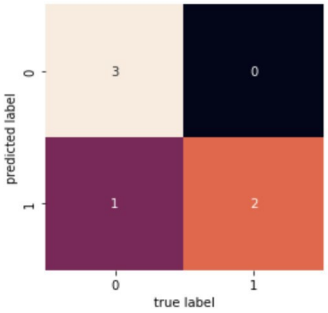

Fig. 10: Confusion Matrix for Decision Tree and Random Forest

CNN results

The loss function evaluates the performance of the model and checks its ability to correctly predict the outcome. It measures the difference between the predicted output and the actual output. If the loss function is reduced, it suggests that the model is improving and making more accurate predictions. The loss function is monitored during the training process and the performance of the model can be accessed, which indicates the improvement of its accuracy.

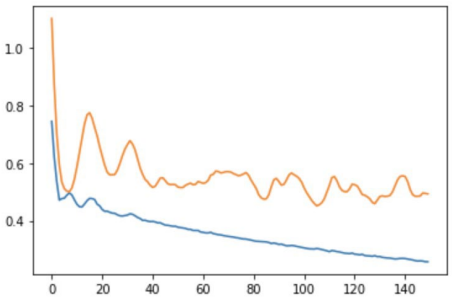

Fig. 11: CNN loss function

Accuracy

Below is a table showing the accuracy of all of the machine learning algorithms.

Classifier	**Accuracy**
Decision Tree	83.3%

Random Forest	83.3%
Convolutional Neural Network	100.0%
Multi-layer perceptron	100.0%

Discussion and Conclusion

The study used machine learning methods like random forest, linear regression, convolutional neural network (CNN), and multilayer perceptron(MLP) to evaluate the use of ECG- derived respiratory signal (EDR) in predicting myocardial infarction (MI). The study's findings indicated that MI can be accurately predicted using features taken from EDR, with deep learning models having the greatest accuracy. This advances the development of MI novel prediction techniques using non-invasive tools like ECG that may eventually be used in biomedical devices.

It should be mentioned that the dataset used only included 20 ECG signals, which is regarded as a small sample size (10 healthy signals and 10 patient signals with MI). In order to further confirm the effectiveness of EDR in predicting MI, future works may benefit the use of more ECG signals. Additionally, working with more machine learning algorithms may enhance the prediction accuracy of MI using EDR. Nonetheless, the results of the study give valuable insights into the potential application of EDR in MI prediction, which paves the way for more research in this area.

Acknowledgement

I would like to express my gratitude to Dr. Shadi Ghiasi for serving as my mentor and guide. Her insight and expertise provided me with invaluable advice, which allowed me to successfully complete this study.

References

[1] Najarian, K., & Splinter, R. (2016). *Biomedical Signal and Image Processing*. https://doi.org/10.1201/b11978
[2] Rangayyan, R. M. (2015). *Biomedical Signal Analysis*. doi:10.1002/9781119068129
[3] deChazal, P., O'Dwyer, M., & Reilly, R. B. (2004). Automatic classification of heartbeats using ECG morphology and heartbeat interval features. *IEEE Transactions on Biomedical Engineering*, *51*(7), 1196–1206. doi:10.1109/tbme.2004.827359
[4] Khan Mamun, M. M., & Alouani, A. (2022). Automatic detection of heart diseases using biomedical signals: A literature review of current status and Limitations. *Lecture Notes in Networks and Systems*, 420–440. doi:10.1007/978-3-030-98015-3_29
[5] Wiharto, W., Kusnanto, H., & Herianto, H. (2016). Intelligence System for diagnosis level of coronary heart disease with K-Star algorithm. *Healthcare Informatics Research*, *22*(1), 30. doi:10.4258/hir.2016.22.1.30
[6] Bashir, S., Qamar, U., & Khan, F. H. (2015). A multicriteria weighted vote-based classifier ensemble for heart disease prediction. *Computational Intelligence*, *32*(4), 615–645. doi:10.1111/coin.12070
[7] Hancock, E. W., Deal, B. J., Mirvis, D. M., Okin, P., Kligfield, P., & Gettes, L. S. (2009). AHA/ACCF/hrs recommendations for the standardization and interpretation of the electrocardiogram. *Circulation*, *119*(10). doi:10.1161/circulationaha.108.191097
[8] Das, M. K., Khan, B., Jacob, S., Kumar, A., & Mahenthiran, J. (2006). Significance of a fragmented QRS complex versus a Q wave in patients with coronary artery disease. *Circulation*, *113*(21), 2495–2501. doi:10.1161/circulationaha.105.595892
[9] Arunachalam, S. P., & Brown, L. F. (2009). Real-time estimation of the ECG-derived respiration (EDR) signal using a new algorithm for Baseline Wander Noise Removal. *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. doi:10.1109/iembs.2009.5333113

[10] Mazzanti, B., Lamberti, C., & de Bie, J. (2003a). Validation of an ECG-derived respiration monitoring method. *Computers in Cardiology, 2003*. doi:10.1109/cic.2003.1291230
[11] Kher, R. (2019). Signal Processing Techniques for Removing Noise from ECG Signals. Journal of Biomedical Engineering and Research, vol 3: 101. doi:10.17303/jber.2019.3.101
[12] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., … SciPy 1.0 Contributors (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2
[13] Shinar, Z., Baharav, A., & Akselrod, S. (n.d.). R wave duration as a measure of body position changes during sleep. *Computers in Cardiology 1999. Vol.26 (Cat. No.99CH37004)*, *26*. doi:10.1109/cic.1999.825903
[14] Sarkar, S., Bhattacherjee, S., & Pal, S. (2015). Extraction of respiration signal from ECG for respiratory rate estimation. *Michael Faraday IET International Summit 2015*. doi:10.1049/cp.2015.1654
[15] Welch, P. (1967). The use of fast fourier transform for the estimation of Power Spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, *15*(2), 70–73. doi:10.1109/tau.1967.1161901
[16] Noh, Y. S., Park, S. B., Hong, K. S., Yoon, Y. R., & Yoon, H. R. (n.d.). A study of significant data classification between EDR extracted and frequency analysis of heart rate variability from ECG using conductive textile. *World Congress on Medical Physics and Biomedical Engineering 2006*, 4100–4103. doi:10.1007/978-3-540-36841-0_1039
[17] Lakdawala, M.M. (2008). Derivation of the respiratory rate signal from a single lead ECG. Retrieved from http://archives.njit.edu/vol01/etd/2000s/2008/njit-etd2008-084/njit-etd2008-084.pdf
[18] Yang, N., Li, T., & Song, J. (2007). Construction of decision trees based entropy and rough sets under tolerance relation. *Proceedings on Intelligent Systems and Knowledge Engineering (ISKE2007)*. doi:10.2991/iske.2007.258
[19] RADU, M. D., COSTEA, I. M., & STAN, V. A. (2020). Automatic traffic sign recognition artificial inteligence - deep learning algorithm. *2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. doi:10.1109/ecai50035.2020.9223186
[20] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. doi: https://doi.org/10.48550/arXiv.1412.6980
[21] Bahi, M., & Batouche, M. (2018). Deep learning for ligand-based virtual screening in Drug Discovery. *2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)*. doi:10.1109/pais.2018.8598488
[22] Delashmit, W. H., & Manry, M. T. (2005, May). Recent developments in multilayer perceptron neural networks. In *Proceedings of the seventh Annual Memphis Area Engineering and Science Conference, MAESC*. Retrieved from https://www.semanticscholar.org/paper/Recent-Developments-in-Multilayer-Perceptron-Neural-Delashmit-Missiles/8657cb338897d912bc417fe3cee7b3ca43a83609
[23] Patel, H. H., & Prajapati, P. (2018). Study and analysis of decision tree based classification algorithms. *International Journal of Computer Sciences and Engineering*, *6*(10), 74–78. doi:10.26438/ijcse/v6i10.7478