POLITECNICO DI TORINO

Master Degree
in Biomedical Engineering

Master Thesis

A hybrid machine learning framework for single-lead ECG signal quality assessment

Supervisor
Dr. Luigi Borzì

Candidate
Giorgio Martorano

Academic Year 2023-2024

Ringraziamenti

Prima di iniziare, desidero esprimere la mia profonda soddisfazione non solo per il raggiungimento di questo importante traguardo accademico, ma soprattutto per il percorso di crescita personale che ha accompagnato questa esperienza. Durante questo viaggio, ho avuto l'opportunità di affrontare sfide che mi hanno permesso di maturare come studente e, ancor più, come persona.

Nulla di tutto ciò sarebbe stato possibile senza il sostegno del mio relatore, il Professor **Luigi Borzì**, al quale desidero esprimere la mia più sincera gratitudine. La sua guida e il suo supporto costante durante tutto il percorso di stesura di questa tesi, insieme alle sue competenze, alla sua disponibilità e ai suoi preziosi consigli, hanno contribuito in modo determinante alla realizzazione di questo lavoro. È stato sempre pronto a rispondere a qualsiasi chiarimento, fornendo spunti di riflessione che hanno arricchito la mia ricerca e migliorato la qualità di questo elaborato.

Senza la sua supervisione, questo traguardo sarebbe stato molto più difficile da raggiungere. Grazie di cuore per avermi accompagnato in questo viaggio di crescita personale e accademica.

Chapter 1

Abstract

Wearable devices are increasingly being used to monitor electrocardiogram (ECG) signals in real time, enabling earlier diagnosis and more effective monitoring of heart health. However, single-lead ECG signals captured by these devices are often contaminated with noise and artifacts, which can degrade signal quality and compromise the accuracy of subsequent diagnostic steps. To prevent this, it's essential to assess the quality of ECG signals before further processing. Manual evaluation of signal quality can be laborious and prone to human error, especially in continuous monitoring contexts with large volumes of data. As a result, developing a streamlined approach for classifying ECG signal quality is critical to improve clinical workflows, reducing human error, and ensuring that diagnostic algorithms receive high-quality input for accurate and timely health assessments. This thesis focuses on developing a robust and efficient system for the automatic classification of single-lead ECG signals based on their quality.

The datasets used in this work included publicly available single-lead ECG signals, each pre-processed to ensure consistency across various data acquisition environments. Pre-processing steps were applied separately to each dataset to ensure that data integrity was preserved. Additionally, each trace was handled carefully to ensure that no data was shared between the training, validation, and test sets, thereby maintaining the reliability of the performance evaluation.

The primary goal of this study is to develop a system capable of accurately distinguishing between high-quality, borderline, and unacceptable ECG signals in single-lead settings. To this end, the study employs a convolutional neural network (CNN) model for initial quality classification, combined with a random forest (RaF) algorithm to refine the classification of borderline signals. The CNN model was enhanced using Generative Adversarial Network (GAN)-based data augmentation to balance the dataset and improve generalization, with traditional data augmentation also evaluated.

The CNN model, when trained using GAN-based data augmentation, achieved an accuracy of 90%, an Area under the curve (AUC) score of 96% and a Recall of 97% on the test set. The RaF classifier, used to further enhance the classification of borderline signals, demonstrated a validation accuracy of 86% and an AUC score of 94%. The cascade of CNN and RaF models performance indicated robust results, particularly for high-quality signals, where the model achieved a precision of 91%, recall of 86%, and F1-score of 88%, resulting in an overall accuracy of 89% with near-perfect recall (99.9%) for unacceptable signals. In addition to classification performance, the computational efficiency of the proposed method was evaluated. The total processing time for a single 5-second ECG signal was approximately 0.2458 seconds, requiring only 1.42 MMACs and

2.89 MFLOPs.

This thesis aims to develop a 3-class classification system for the quality of the ECG signal that is very energy efficient and therefore with the possibility of being applied in real-time monitoring systems. Future work will focus on optimizing the model for portable wearable systems, focusing on energy and computational efficiency.

Contents

Ringraziamenti 1

1 Abstract 2

List of Tables 6

List of Figures 7

2 Introduction 9
2.1 Fundamental Properties of ECG Signals . . . 10
2.1.1 Core Features of ECG Signal . . . 10
2.1.2 Typical Artifacts in ECG Signals . . . 11
2.2 Thesis Objective and Organization . . . 12

3 Background 14
3.1 Traditional Methods of ECG-SQA . . . 14
3.1.1 Manual Visual Analysis . . . 14
3.1.2 Statistical Signal Quality Indices . . . 14
3.2 Advanced Methods for ECG-SQA . . . 19
3.2.1 Machine Learning Signal Quality Indices . . . 19
3.2.2 Deep Learning Methods . . . 24
3.2.3 Introduction to GANs . . . 30
3.3 Limitations of Prior Research and Study Rationale . . . 34

4 Materials and Methods 36
4.1 Dataset . . . 36
4.2 Pre-Processing . . . 38
4.2.1 Summary of Preprocessing Steps . . . 38
4.2.2 Datasets handling . . . 38
4.2.3 Scaling . . . 39
4.2.4 Downsampling . . . 39
4.2.5 Segmentation . . . 41
4.2.6 Annotator Processing . . . 41
4.2.7 Dataset Construction . . . 42
4.2.8 Data Augmentation . . . 44
4.2.9 Traditional Data Augmentation . . . 45
4.2.10 WGAN-GP Data Augmentation . . . 46
4.2.11 Training Process . . . 48
4.2.12 Band-Pass Filtering . . . 49

4.3 Classification Algorithm . . . 50
4.3.1 CNN . . . 50
4.3.2 Random Forest (RaF) Classifier . . . 51
4.3.3 CNN Training and validation . . . 53
4.3.4 Performance Evaluation . . . 54
4.3.5 Algorithm Complexity Analysis . . . 56

5 Results 57
5.1 CNN Results . . . 57
5.1.1 Traditional Data Augmentation Results . . . 57
5.1.2 GAN-Based Data Augmentation Results . . . 58
5.2 RaF Results . . . 59
5.3 Final Cascade Model Results . . . 60
5.4 Algorithm Complexity Results . . . 61

6 Discussion 63
6.1 CNN Performance . . . 63
6.1.1 Comparison of Deep Learning-Based ECG Classification Methods 63
6.2 RaF Performance . . . 66
6.3 Cascade Model Performance . . . 67
6.4 Algorithm Complexity . . . 68
6.5 Limitations . . . 68

7 Conclusion and Future Work 70
7.1 Conclusion . . . 70
7.2 Future Work . . . 70

List of Tables

2.1 Typical frequency ranges for different ECG wave components [64]. 11
3.1 Maximum accuracy of SQIkur, SQIp, SQIsnr and SQIhos for dataset ECG-ID, Tele ECG, BIDMC, MIT/BIH arrhythmia, CINC 2011 and CINC 2014. 17
3.2 Single-lead classification using individual SQIs best performing SQI indicator is shown in bold and underlined 17
3.3 Meanings of the four "flags" 18
3.4 Confusion matrix summarizing classification results 18
3.5 Confusion matrix summarizing classification results in horse activity recognition using textile electrodes. 18
3.6 Classifier accuracy. Note that for voting, the results are simply from the majority vote of the SVM, MLP and LDA classifiers. ‡ indicates balanced data. ♯ indicates updated annotations. Best results are underlined. . . . 22
3.7 Highest Accuracy Achieved by Each Classifier on Test Data 22
3.8 Performance Metrics for ECG Signal Quality Classification. 24
3.9 Summary of Deep Learning Related Works with Preprocessing and Model Details. 30
4.1 Summary of Datasets Used in the Study 38
4.2 Downsampling Error Comparison (100Hz vs 200Hz) in Percentages . . . 40
4.3 Distribution of Signals Across Datasets for Each Label (CNN) 43
4.4 Final label distribution in Training, Validation, and Test sets. 43
4.5 Total Distribution of High-Quality and Borderline Signals in QDB Dataset for Random Forest Model 44
4.6 Distribution of High-Quality and Borderline Signals in Training and Validation Sets 44
4.7 Final label distribution in Training, Validation and Test sets. 46
4.8 Final label distribution in Training, Validation, and Test sets. The Training set includes an additional 20,000 synthetic signals generated by the WGAN model and 5,000 WGN signals to balance the dataset. 49
5.1 Evaluation Metrics for CNN Model with Traditional Data Augmentation 58
5.2 Evaluation Metrics for CNN Model with GAN-Based Data Augmentation 58
5.3 RaF Model Evaluation Metrics on Training and Validation Sets 60
5.4 Class-wise Metrics and Overall Accuracy 61
5.5 Computational time for each phase of the data processing pipeline. . . . 61
6.1 Summary of Deep Learning Related Works with Preprocessing and Model Details. 64

List of Figures

2.1 The morphological characteristics of the ECG signal, including the S-T segment, QRS complex, P-R and Q-T intervals. From [8] 10
2.2 Baseline wander in ECG signal. From [50]. 11
2.3 Power line interference in ECG signal. From [50]. 12
2.4 Muscle artifact in ECG signal. From [50]. 12
2.5 Electrode Motion in ECG signal. From [50]. 12
3.1 Random Forest Simplified: An example of how multiple decision trees work together in a random forest to classify an instance through majority voting [33]. 20
3.2 1D-CNN model with two convolutional and max pooling layers feeding a dense fully connected layer [74]. 25
3.3 The structure of a GAN [73] 32
3.4 Comparison of Gradient Norms and Weights for Weight Clipping and Gradient Penalty. This shows the behavior of gradient norms across discriminator layers using weight clipping with different clipping values and gradient penalty. From [55]. 34
4.1 Overview of the ECG signal preprocessing steps. The green boxes represent steps common to all datasets, while the yellow box indicates a step exclusive to the QDB dataset. 38
4.2 Example of transformations applied during data augmentation. The original ECG signal (top) undergoes time stretching, pitch shifting, noise addition, and amplitude modification. 46
4.3 Critic Network Architecture. 47
4.4 Generator Network Architecture. 47
4.5 Comparison of Real and Generated ECG Signals. The top row shows the real ECG signals, while the bottom row shows the corresponding generated signals from the WGAN model. 49
4.6 Frequency response of the Chebyshev Type I bandpass filter of order 5 used in preprocessing the ECG signals. The filter passes frequencies between 0.8 Hz and 40 Hz, removing high-frequency noise and low-frequency baseline wander. 50
4.7 Flowchart depicting the classification algorithm. The ECG signals are preprocessed and passed through a CNN model and a RaF model in sequence, leading to a final classification into three labels (0, 1, 2). Each classifier is trained on different datasets, as shown. 50
4.8 Architecture of the CNN model used for ECG signal classification. The network comprises five 1D convolutional layers with increasing filters, followed by pooling, dropout, and dense layers. LeakyReLU activations are applied after each convolutional and dense layer. 51

4.9 Comparison of SQI metrics for high-quality (blue) and borderline signals (red), for both training and validation sets. 53
4.10 Training Loss vs Validation Loss during training with smoothed moving averages. The smoothed curves help visualize the trend and stability of loss over epochs. 55
5.1 ROC Curves for Training, Validation, and Test Sets with Traditional Data Augmentation 57
5.2 ROC Curves for Training, Validation, and Test Sets with GAN-Based Data Augmentation 58
5.3 Training Set Confusion Matrix for Random Forest 59
5.4 Validation Set Confusion Matrix for Random Forest 60
5.5 Confusion Matrix of the Final Cascade Model on the Validation Set . . . 61
6.1 Examples of False Negatives and False Positives in ECG Signal Classification. 67

Chapter 2

Introduction

Cardiovascular diseases (CVDs) are the leading cause of mortality worldwide, resulting in approximately 17.9 million deaths in 2019 [48]. Early detection and monitoring of heart-related conditions are crucial in preventing fatal outcomes. Electrocardiograms (ECGs) play a vital role in this context by providing non-invasive and continuous monitoring of heart electrical activity. ECGs are widely used for diagnosing various cardiac abnormalities, including arrhythmias, myocardial infarctions, and other heart-related conditions [78].

However, the accuracy of ECG-based diagnostics heavily depends on the quality of recorded signals. Poor-quality ECG signals, often contaminated by noise, motion artifacts, or improper electrode placement, can lead to incorrect diagnoses and ineffective treatment plans [60]. In clinical and remote monitoring environments, ensuring high-quality ECG signals is paramount, especially with the increasing use of wearable devices.

Traditionally, ECG signal quality assessment (ECG-SQA) was carried out by experienced clinicians who visually inspected the signal traces and classified them according to their experience. Although this may be considered a reliable method, it is very time-consuming and extremely subjective [69]. For this reason, interest in automated classification systems, capable of handling large amounts of data, has been growing in recent years. These methods range from statistical approaches to machine learning and deep learning approaches [38].

In recent years, deep learning has emerged as a powerful tool for analyzing complex biomedical signals, including ECGs [26]. Deep learning models, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have demonstrated remarkable success in automatically extracting features from ECG signals and classifying them into different categories [51]. By leveraging large datasets and sophisticated architectures, deep learning models can learn to distinguish between acceptable and unacceptable ECG signals with high accuracy, thereby reducing the burden on clinicians and improving the reliability of remote monitoring systems.

Although the results proposed so far are promising, many issues still need to be addressed, such as the high variability of ECG signals acquired with different systems or the training of models that can be used in real-time applications where energy and computing resources are limited [52]. Addressing these challenges requires the development of robust deep learning systems as well as efficient data manipulation.

2.1 Fundamental Properties of ECG Signals

2.1.1 Core Features of ECG Signal

The ECG is a non-invasive diagnostic tool that records the electrical activity of the heart over time. The heart's rhythmic contractions and relaxations are controlled by electrical impulses generated in the sinoatrial (SA) node, which serves as the heart's natural pacemaker. These electrical signals propagate through the heart muscles, resulting in the contraction of atria and ventricles, which pumps blood throughout the body [59]. The ECG captures these signals using electrodes placed at specific locations on the patient's body, typically on the chest and limbs, providing a graphical representation of the heart's electrical activity over time [49].

An ECG signal consists of several key components, including the P wave, QRS complex, and T wave, each corresponding to specific phases of the cardiac cycle. The P wave represents the depolarization of the atria, the QRS complex corresponds to the rapid depolarization of the ventricles, and the T wave signifies ventricular repolarization [25]. In addition, various intervals and segments such as the P-R interval, S-T segment, and Q-T interval provide critical diagnostic information for detecting and classifying arrhythmias and other cardiovascular conditions, as shown in Figure 2.1.

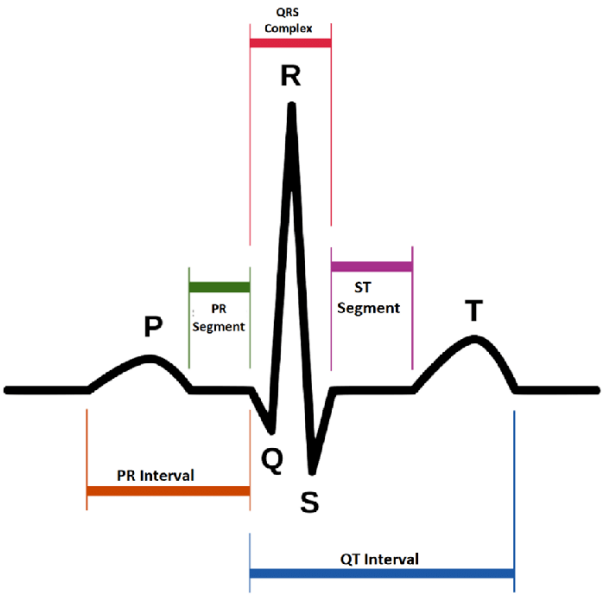


Figure 2.1: The morphological characteristics of the ECG signal, including the S-T segment, QRS complex, P-R and Q-T intervals. From [8]

.

The distinct components of the ECG signal are characterized by unique spectral signatures. These frequency ranges are crucial for deciphering the various phases of the cardiac cycle and identifying potential anomalies. The typical frequency ranges associated with the P wave, QRS complex, and T wave, are presented in the Table 2.1 below.

Table 2.1: Typical frequency ranges for different ECG wave components [64].

ECG Component	Frequency Range (Hz)
P wave	5 - 30
QRS complex	8 - 50
T wave	0 - 10

2.1.2 Typical Artifacts in ECG Signals

While the ECG is a powerful diagnostic tool, it is susceptible to various types of noise and artifacts, which can significantly distort the recorded signal and hinder accurate diagnosis [10]. Understanding the characteristics and sources of these noise components is critical for developing effective pre-processing techniques in ECG signal analysis.

Baseline Wander (BW)

Low-frequency noise in ECG signals, also known as baseline wander, can be attributed to various factors. These include respiratory movements, bodily movements, poor electrode contact, and changes in skin-electrode impedance [21]. In some cases, the drift in the baseline can be substantial, reaching up to 15% of the full-scale deflection. This type of noise typically occurs within a frequency range of 0.15-0.30 Hz [13]. An example of BW is reported in Figure 2.2

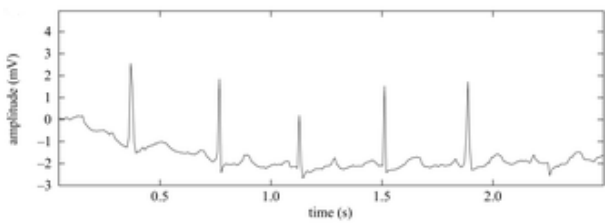


Figure 2.2: Baseline wander in ECG signal. From [50].

Moreover, abnormal breathing rates and electrode movement can significantly exacerbate the problem, introducing additional motion artifacts that can further distort ECG features such as the ST-segment. This can lead to misdiagnosis of various conditions, including myocardial infarction, Brugada syndrome, and other ST-segment-related abnormalities [56].

Power Line Interference (PLI)

One of the primary sources of noise disturbance in ECG signals is the influence of electromagnetic fields emanating from power infrastructure. This type of interference can be amplified by poor grounding of the recording device or the patient's electrical environment. The resulting distortion, as illustrated in Figure 2.3, appears as a repetitive pattern at a frequency corresponding to the primary power grid frequency (typically 50 or 60 Hz). This disturbance can lead to compromised visibility of key ECG features, such as P-waves, thereby increasing the risk of misinterpretation of atrial arrhythmias [13].

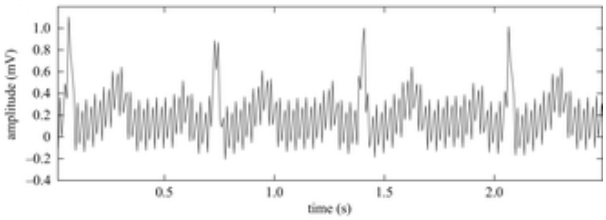


Figure 2.3: Power line interference in ECG signal. From [50].

Muscle Artifact

Electrical signals emanating from skeletal muscle activity can contaminate ECG recordings, causing artifacts that are frequently mistaken for genuine ECG signals. These artifacts arise when muscle electrical activity coincides with the recording process, typically during muscle movement or tensing, particularly in areas proximal to electrode placement sites. The frequency spectrum of muscle artifacts often overlaps with that of the ECG signal, making it intricate to distinguish between the two. As a consequence, these artifacts can modify key ECG features leading to inaccurate interpretation of cardiac electrical activity [13]. An example of such a muscle artifact is shown in Figure 2.4 .

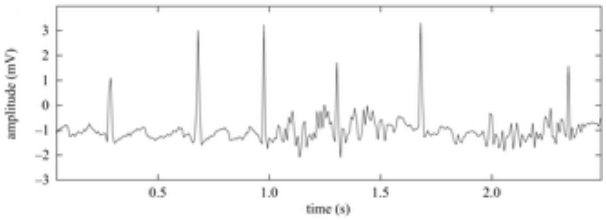


Figure 2.4: Muscle artifact in ECG signal. From [50].

Electrode Motion Artifacts

Unstable electrode connections can introduce distortion into the ECG signal, often due to issues with electrode attachment or the interaction between the skin and the electrode [50]. As shown in Figure 2.5, this can lead to the distortion of characteristic peaks of the P, Q, R, S, and T-waves, making it challenging to accurately interpret the ECG reading.

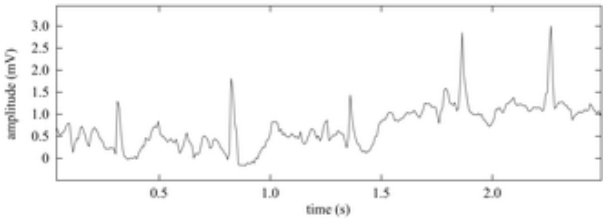


Figure 2.5: Electrode Motion in ECG signal. From [50].

2.2 Thesis Objective and Organization

The primary objective of this thesis is to develop a robust and efficient system for classifying ECG signals into three quality categories using deep learning and ML methods. The

focus is on leveraging advanced ML techniques to improve the accuracy and reliability of ECG-SQA in wearable devices. The study aims to address the challenges posed by noisy and artifact-ridden ECG signals, which can lead to inefficient analyses, increased power consumption, and potentially inaccurate diagnostics and features calculation. By employing a CNN model enhanced with GAN-based data augmentation, and integrating a RaF classifier, the thesis seeks to create a system that can accurately classify ECG signals, ensuring that borderline-quality signals undergo more intensive processing for further analysis, while high-quality signals are processed minimally, and unacceptable signals are discarded. This not only enhances the reliability of ECG-based monitoring systems but also conserves battery life in wearable devices and prevents false diagnostic outcomes.

The structure of the work, divided into chapters, is as follows:

- **Chapter 2: Background** – This chapter reviews the existing methods for ECG-SQA, with a focus on both traditional and advanced ML techniques. It discusses the evolution of these methods and their applicability to real-time ECG analysis.
- **Chapter 3: Materials and Methods** – This chapter details the datasets used in the study, the preprocessing techniques applied to the ECG signals, and the design of the CNN and RaF models. It also explains the data augmentation strategies and the process of model training and validation.
- **Chapter 4: Results** – This chapter presents the performance of the CNN and RaF models, including the results of the final cascade model. It compares the effectiveness of traditional and GAN-based data augmentation and discusses the computational complexity of the proposed methods.
- **Chapter 5: Discussion** – This chapter analyzes the results, highlighting the strengths and limitations of the models. It compares the findings with those of previous studies and discusses the implications for real-world applications.
- **Chapter 6: Conclusion and Future Work** – This chapter summarizes the key contributions of the thesis, identifies the limitations, and outlines directions for future research, including the integration of additional datasets, exploration of advanced data augmentation techniques, and optimization of the models for deployment on wearable devices.

Chapter 3

Background

Over the past few decades, significant progress has been made in the research of ECG signal analysis, leading to improved accuracy and efficiency in cardiac diagnosis. This chapter aims to review a wide range of studies related to ECG-SQA, tracing its evolution and focusing on applications based on deep learning methods. Particular emphasis will be placed on studies that have utilized deep learning algorithms, such as CNNs, possibly supported by traditional data augmentation techniques and Generative Adversarial Networks (GANs).

The chapter will first cover traditional approaches to ECG signal analysis and then explore recent research introducing Artificial Intelligence (AI), highlighting a shift towards more automated and objective techniques.

3.1 Traditional Methods of ECG-SQA

3.1.1 Manual Visual Analysis

Manual visual analysis is undoubtedly one of the most reliable techniques for assessing the quality of an ECG signal. This method relies on the expertise of a specialist who, through visual inspection of the ECG trace, can classify the signal as acceptable or unacceptable. Although this method is highly reliable and robust, it has significant limitations, such as being time-consuming and unsuitable for analyzing large volumes of data. Additionally, signal interpretation can vary from one expert to another, introducing a subjective component. For these reasons, research is moving towards the development of automated solutions that, while not always reaching the reliability level of manual analysis, offer significant advantages in terms of efficiency and objectivity in evaluations.

3.1.2 Statistical Signal Quality Indices

Statistical analysis is a cornerstone method for assessing the overall quality of the signal, with a particular focus on identifying the presence of noise or artifacts. Four key indices commonly utilized for this purpose are the **relative power of the QRS complex (SQIp)**, **skewness (SQIskew)**, **signal-to-noise ratio (SQIsnr)**, and **kurtosis (SQIkur)**. These metrics enable a quantitative assessment of signal quality, allowing for streamlined automation and efficient analysis.

Kurtosis

The kurtosis coefficient is used to examine the disparities between datasets with centrally clustered values and those with more anomalous values. This coefficient demonstrates the degree of truncation or peakiness in a distribution. A higher kurtosis showcases the presence of more prominent outliers, whereas a lower kurtosis denotes the absence of such anomalies [31].

According to previous research by Zaho, the SQIkur metric can serve as a significant ECG signal quality indicator. Signals exhibiting poor quality are characterised by low kurtosis due to the corrupting influence of noise and artifacts, resulting in a more uniform distribution with fewer distinct peaks. Conversely, high-quality ECG signals tend to display higher kurtosis as they often feature prominent QRS complexes producing more pronounced peaks in the distribution [77].

SQIkur can be calculated using the following formula:

$$\text{SQIkur} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \bar{x}}{\sigma}\right)^4, \tag{3.1}$$

where x denotes the ECG signal consisting of N sample points, $\bar{x}$ signifies the mean value of the signal x, and σ denotes the standard deviation of the signal x.

Signal-to-noise ratio

The SQI Signal-to-Noise-Ratio (**SQIsnr**) is a metric that quantifies the signal-to-noise ratio by comparing the variability of the ECG signal to the variability of the noise present in the signal. The signal diversity represents the variance of the absolute ECG signal amplitude, whereas the noise diversity is defined as the variance of the signal itself [19].

The SQIsnr can be estimated using the following formula:

$$\text{SQIsnr} = \frac{\sigma_y^2}{\sigma_{|y|}^2}, \tag{3.2}$$

where y represents the ECG signal.

Skewness

In statistics, skewness examines the degree of asymmetry in a probability distribution. A distribution is characterized as skew when its left and right halves do not possess reflective symmetry. Distributions can exhibit three types of skewness: positive, negative, or neutral. A positively skewed distribution tends to have a longer tail on the right side of its central tendency, whereas a negatively skewed distribution displays a longer tail on the left side [67]. The concept of skewness can adapt to distinct patterns in noise, demonstrating flexibility depending on the type of noise present. In specific instances, high-frequency noise might generate a symmetrical distribution, accompanied by a skewedness value that is relatively low. This observation highlights the limitations of relying solely on skewness to characterize signals [77]. Given these limitations of skewness in accurately assessing ECG signal quality, it is beneficial to consider a combined metric that leverages the strengths of both skewness and kurtosis.

Higher-Order-Statistics

Nardelli et al. [46] introduced an innovative method for assessing the quality of ECG signals, which they called Higher-Order-Statistics SQI (SQIhos). By combining the strengths of two existing metrics, SQIhos offers a robust approach to evaluating signal quality .

The formula for SQIhos is a weighted combination of skewness and kurtosis. Specifically, it is defined as:

$$\text{SQIhos} = |\text{SQIskew}| \times \frac{\text{SQIkur}}{5}, \tag{3.3}$$

where **SQIskew** is denoted by:

$$\text{SQIskew} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \bar{x}}{\sigma}\right)^3. \tag{3.4}$$

Relative power

The ECG signal is comprised of various components, including the P wave, QRS complex, and T wave, among others. The QRS complex is the most significant component, accounting for approximately 99% of the signal's energy. It is characterized by a distinctive frequency band centered at 10 Hz, with a bandwidth of 10 Hz [77].

Spectral analysis of the ECG signal enables the calculation of the power spectral densities (PSDs) of the signal and its QRS complex. The ratio of these two PSDs, denoted by SQIp, is a novel index that provides insights into the quality of the ECG signal. The SQIp is calculated as the ratio of the energy contained in the QRS complex frequency band to the total energy of the ECG signal ($P(f)$).

$$\text{SQIp} = \frac{\int_{5,\text{Hz}}^{15,\text{Hz}} P(f), \mathrm{d}f}{\int_{5,\text{Hz}}^{45,\text{Hz}} P(f), \mathrm{d}f}, \tag{3.5}$$

In the presence of electromyographic (EMG) interference, the high-frequency component of the ECG signal increases, leading to a decrease in SQIp. Thus, this index can be used as an effective criterion for identifying the presence of EMG interference in ECG signals [77].

Applications

Rahman et al. (2022) [50] investigated the quality of ECG signals by examining the performance of various metrics and different segmentation lengths in seconds (1 , 2, 5, 10). In addition, the authors developed a graphical user interface (GUI) for manually labelling the segments. They extracted a range of features from pre-processed ECG signals, including the relative power of the QRS complex, kurtosis, signal-to-noise ratio, and skewness. A statistical analysis was employed to compare these metrics.

The proposed method was tested on a diverse set of six datasets, namely the MIT-BIH, ECG-ID, Tele ECG, BIDMC, PhysioNet/CinC Challenge 2011 (CINC 2011), and PhysioNet/CinC Challenge 2014 (CINC 2014) datasets, using pre-specified window sizes. The results showed that the overall accuracy for each SSQI changes extremely across the datasets as shown in Table 3.1. This indicates that the effectiveness of each SSQI is highly dependent on the specific characteristics of the dataset being analyzed.

Table 3.1: Maximum accuracy of SQIkur, SQIp, SQIsnr and SQIhos for dataset ECG-ID, Tele ECG, BIDMC, MIT/BIH arrhythmia, CINC 2011 and CINC 2014.

QI	window size/Acc.	dataset					
		ECG-ID	Tele ECG	BIDMC	MIT/BIH	CINC 2011	CINC 2014
SQIkur	window size (Sec.)	1	1	10	2	2	5
	Acc. (%)	77.96	57.48	83.95	89.58	72.33	81.23
SQIp	window size (Sec.)	2	10	10	10	2	5
	Acc. (%)	66.25	59.58	60.85	52.91	62.55	70.14
SQIsnr	window size (Sec.)	1	5	5	2	1	1
	Acc. (%)	66.73	73.40	84.43	97.51	57.01	80.89
SQIhos	window size (Sec.)	2	1	2	2	1	10
	Acc. (%)	74.81	56.81	81.00	84.64	77.2	76.38

Zhao and Zhang (2018) [77] developed an effective system for assessing the quality of ECG signals using a combination of simple heuristic fusion and fuzzy comprehensive evaluation of signal quality indices (SQIs). The authors aimed to develop a robust system for ECG quality assessment that combines simple heuristic fusion and fuzzy comprehensive evaluation. Six SQIs were extracted and quantified based on noise characteristics and ECG waveform features: R peak detection match (qSQI), QRS wave power spectrum distribution (pSQI), Kurtosis (kSQI), Baseline relative power (basSQI), Variability in the R-R interval, and Skewness.

To evaluate the SQIs, the researchers employed a fuzzy comprehensive method that incorporated Cauchy, rectangular, and trapezoidal distributions to calculate the membership functions. A fuzzy vector was established, and the bounded operator was utilized for fuzzy synthesis, with weighted membership functions used for assessment and classification.

The study used two datasets: PhysioNet/CinC Single-lead Challenge 2017 (PCCC 2017, D1) and PhysioNet/CinC Challenge 2011 (PCCC 2011, D2). From the latter, only lead II was extracted. To ensure the reliability and accuracy of the results, a 10-fold cross-validation method was employed, demonstrating high sensitivity, accuracy, and specificity in ECG-SQA. A summary of the performance metrics for both datasets is provided in Table 3.2.

Table 3.2: Single-lead classification using individual SQIs best performing SQI indicator is shown in bold and underlined

	qSQI	**pSQI**	**cSQI**	**kSQI**	**basSQI**
Database D1					
Acc	**80.33**	**80.00**	76.00	**79.67**	**78.67**
Se	95.33	95.00	63.67	84.33	80.67
Sp	88.33	80.33	56.33	83.00	72.33
Database D2					
Acc	**86.33**	**77.00**	74.33	**82.33**	**83.00**
Se	93.67	84.33	66.33	85.00	86.00
Sp	80.67	69.67	47.67	80.67	79.67

Liu et al. (2011) [39] proposed a novel approach to ECG-SQA using the PCCC 2011 dataset.

The dataset, consisting of standard twelve-lead ECGs recorded for ten seconds, was divided into a training set to design and test the algorithm. The dataset included 773

acceptable ECGs and 225 unacceptable ECGs. The authors introduced four "flags" to identify specific issues that degrade ECG signal quality (Table 3.3). These flags were then combined to calculate an Index of Signal Quality for each ECG lead and an Integrative Signal Quality Index for the twelve-lead ECGs.

Table 3.3: Meanings of the four “flags”

Flags	Meaning for the ECG	Values
Flag1	Straight line or not?	Yes (1) / No (0)
Flag2	Includes a large impulse?	Yes (1) / No (0)
Flag3	Gaussian noise present?	Yes (1) / No (0)
Flag4	Detector error in R-wave peak detection?	Yes (1) / No (0)

The validity of the proposed method was evaluated using two indices: sensitivity and specificity, which were found to be 90.67% and 89.78%, respectively. This demonstrates the effectiveness of the proposed method in assessing ECG signal quality. The main results are summarized in Table 3.4.

Table 3.4: Confusion matrix summarizing classification results

	Predicted Acceptable	Predicted Unacceptable
Actual Acceptable (773)	694 (**N1**)	79 (**N2**)
Actual Unacceptable (225)	21 (**A1**)	204 (**A2**)

Nardelli et al. (2020) [46] developed a novel real-time SQI called SQIhos for evaluating the quality of ECG recordings. The developed SQI, is designed to enhance the performance of existing SQIs by introducing two new indices, SQIkur and SQIskew. The authors validated SQIhos using a dataset of 1000 human twelve-lead ECGs from the PCCC 2011 Challenge dataset and demonstrated its superiority over four existing SQIs, achieving an accuracy of 90.38% in human signal quality discrimination.

Furthermore, the study applied SQIhos to investigate the quality of ECG signals acquired using two different electrode systems, traditional red-dot electrodes and textile electrodes, during submaximal treadmill tests in horses. The analysis revealed that textile electrodes provided significantly better signal quality than red-dot electrodes, particularly in high-motion conditions such as galloping. To distinguish between three activity conditions (walk, trot, and gallop) based on the SQIs, a real-time pattern recognition algorithm using a C-SVM classification model was implemented. The results showed that SQIhos was the most discriminant feature, achieving an accuracy of 84.91% in distinguishing between walking and galloping.

Here is a summary of the classification results in horse activity recognition using textile electrodes (Table 3.5):

Table 3.5: Confusion matrix summarizing classification results in horse activity recognition using textile electrodes.

Activity	Predicted Walk	Predicted Trot	Predicted Gallop
Actual Walk	65.71%	27.18%	7.12%
Actual Trot	28.24%	47.29%	24.47%
Actual Gallop	12.35%	12.47%	75.18%

3.2 Advanced Methods for ECG-SQA

3.2.1 Machine Learning Signal Quality Indices

Machine Learning (ML) is a branch of AI that focuses on the development of algorithms that enable computers to learn from and make decisions based on data [11]. Unlike traditional programming, where explicit instructions are given, ML algorithms identify patterns within data and use these patterns to make predictions or decisions. This ability to learn and adapt makes ML particularly useful for applications where the relationship between input data and output results is complex or not well understood.

In the context of ECG-SQA, ML models are capable of analyzing large amounts of data and identifying patterns characterized by artifacts. By training the ML model with labeled data, it is able to predict new signals that have never been explored before. This overcomes the problems of the previously mentioned statistical threshold methods, which are characterized by poor generalizability.

The most commonly used ML classifiers for signal quality assessment in ECG include:

- **Support Vector Machines (SVM)**: A supervised learning algorithm that finds the hyperplane that best separates data into classes [41].

- **Linear Discriminant Analysis (LDA)**: a method for identifying a linear combination of features that classifies or distinguishes two or more classes of objects or events [2].

- **Multilayer Perceptron (MLP)**: A type of neural network that consists of multiple fully-connected layers of neurons, capable of modeling complex relationships in data [41].

- **Naive Bayes (NB)**: A probabilistic classifier based on Bayes' theorem, assuming independence between the features [1].

- **Random Forest (RaF)**: RaF is a sophisticated supervised learning algorithm that combines multiple decision trees to generate more accurate and robust predictions [12]. By repeatedly sampling the training data and features, RaF mitigates the risk of overfitting and enhances the model's ability to generalize to new, unseen data [28].

  Each individual decision tree in RaF makes a prediction, and the final outcome is calculated by aggregating the predictions. The algorithm's ability to reduce correlation between trees, achieved through random sampling, is a crucial factor in its success. This property enables RaF to outperform a single decision tree in many applications [12]. An example of RaF structure could be seen in Figure 3.1

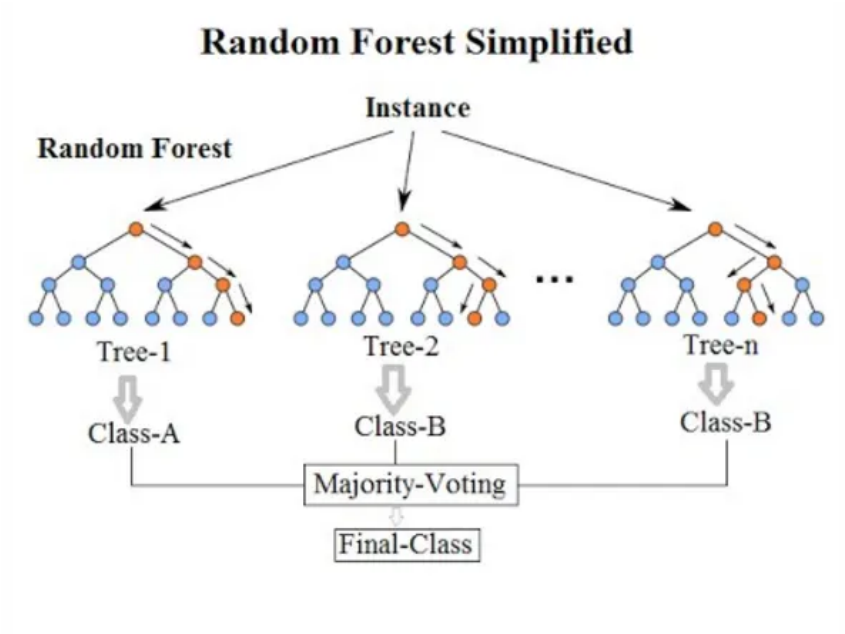


Figure 3.1: Random Forest Simplified: An example of how multiple decision trees work together in a random forest to classify an instance through majority voting [33].

The RaF offers many advantages that make it an excellent candidate for adoption. It can learn non-linear relationships between variables and more importantly it is versatile in the ways it can be used with different types of data. In addition, the Random Forest provides information on the importance of each variable, offering valuable support in the interpretation of decisions made by the model [6].

While helpful in many ways, RaF does have certain drawbacks. This is a computationally expensive process in general, especially for larger datasets. Moreover, since it is very sensitive to noisy data, it often runs the risk of increasing the probability of overfitting [6].

The RaF algorithm is built on the foundation of decision trees. Each decision tree asks a series of yes/no questions about the data, gradually narrowing down the possible outcomes until a final prediction is made. For instance, when forecasting tomorrow's maximum temperature, a decision tree might first inquire about the season. If it's winter, the tree might then refine its prediction by asking additional questions, such as the historical average temperature for that date [33].

RaF takes this concept a step further by generating a large number of decision trees, each trained on a random subset of the data and features. This randomness introduces diversity among the trees, leading to more robust and accurate predictions. By aggregating the predictions from all the trees (in regression tasks) or taking a majority vote (in classification tasks), RaF can outperform individual decision trees, which may be prone to overfitting or biased by specific subsets of the data [33].

The "Random Forest" name reflects the algorithm's reliance on randomness in the selection of data subsets and features for each tree. This randomness ensures that each tree has a unique perspective on the data, contributing to the overall robustness of the model [33]. When the trees are combined, they form a model that is more accurate and generalizable than any single tree could be on its own.

Applications

Clifford et al. 2012 [16] investigated the application of ML techniques for evaluating signal quality in ECG signals. The researchers extracted six SQIs from PCCC 2011 dataset:

- Integrative Signal Quality Index (**iSQI**): measures the percentage of beats detected on each lead that were also detected on all leads.

- bSQI: assesses the percentage of beats detected by two common QRS complex detectors, *eplimited* and *wqrs*.

- Frequency Signal Quality Index (**fSQI**): calculates the ratio of power in the frequency band 5-20 Hz to the total power. The fSQI can be calculated using the following formula:

$$\text{fSQI} = \frac{P(5\text{Hz} - 20\text{Hz})}{P(0\text{Hz} - f_n\text{Hz})} \tag{3.6}$$

  where $P(f)$ represents the power in the frequency band f, and f_n is the Nyquist frequency.

- sSQI

- kSQI.

- pSQI.

The ECG data was downsampled to 125 Hz for each channel using an anti-aliasing filter. The authors employed cross-validation and evaluation metrics such as accuracy, sensitivity, and specificity to assess the performance of the classifiers. The results showed that SVM and MLP neural networks demonstrated exceptional performance, achieving accuracy rates of up to 99% and 95%, respectively, as showed in Table 3.6. The study highlighted the potential of ML to enhance ECG-SQA in noisy environments, which can have significant implications for the field of cardiology.

Table 3.6: Classifier accuracy. Note that for voting, the results are simply from the majority vote of the SVM, MLP and LDA classifiers. ‡ indicates balanced data. ♯ indicates updated annotations. Best results are underlined.

Method ↓	**Train Set-a**	**Test Set-b**	**PNet Score**	**Train Set-b**	**Test Set-a**	**PNet Score**
SVM	1.000	0.950	0.904	1.000	0.934	0.926
SVM‡	0.986	0.932	0.862	1.000	0.885	0.926
MLP	0.990	0.954	0.892	0.992	0.935	0.922
MLP‡	0.996	0.954	0.888	1.000	0.930	<u>0.926</u>
MLP♯	0.978	0.918	0.890	0.992	0.880	<u>0.940</u>
MLP‡♯	0.993	0.928	0.890	1.000	0.876	0.936
NB	0.911	0.936	0.890	0.942	0.907	0.880
NB‡	0.911	0.936	0.890	0.940	0.909	0.894
LDA	0.949	0.942	0.900	0.960	0.921	0.890
LDA‡	0.928	0.910	0.880	0.902	0.897	0.876
VOTE	0.994	0.948	0.902	0.996	0.934	0.922
VOTE‡	0.994	0.942	0.876	1.000	0.933	0.926

Zhang et al. (2019) [76] explored the performance of various ML algorithms for ECG quality assessment. The focus of the study was on both iterative and non-iterative classification models. The researchers evaluated the effectiveness of four classifiers: Kernel-SVM (KSVM), RaF, Least Squares-SVM (LS-SVM), and multi-surface proximal SVM based oblique-RaF (ORaF). These algorithms were tested using seven feature schemes derived from a combination of 27 linear and nonlinear features, including novel features such as Encoding Lempel-Ziv Complexity, Permutation Entropy, and Approximate Entropy.

The dataset utilized in the study was PCCC2011 and consisted of 1,500 mobile ECG recordings collected using smartphones, with 1,000 recordings designated as training data and 500 as test data. These recordings were annotated by clinical experts as either "acceptable" or "unacceptable". Prior to classification, the feature vectors were normalized to a zero-mean standardization.

The results revealed that the inclusion of nonlinear features, particularly Permutation Entropy and Encoding Lempel-Ziv Complexity, together with power spectral features, significantly improved classification performance. Among the classifiers, LS-SVM demonstrated superior accuracy, achieving a classification accuracy of 92.20% on the test dataset when using the sixth feature scheme, which included all features except Approximate Entropy. While RaF showed strong performance on the training data, its generalization ability was found to be limited, as reflected by its lower accuracy on the test data. ORaF performed better in generalization than RaF, but did not surpass LS-SVM.

Table 3.7: Highest Accuracy Achieved by Each Classifier on Test Data

Classifier	**Accuracy (%)**	**Feature Scheme**
LS-SVM	92.20	Scheme 6 (Waveform + Frequency + PE + ELZC)
KSVM	92.00	Scheme 6
RaF	91.40	Scheme 6
ORaF	92.00	Scheme 6

Kužílek et al. (2011) [34] developed an innovative algorithm for assessing the quality

of ECG signals obtained using mobile phones, as a response to the PCCC 2011. Their method seamlessly integrates straightforward rules to eliminate low-quality recordings (e.g., high-amplitude noise, detached electrodes) with a sophisticated SVM classification for more complex cases.

The algorithm consists of three stages: (1) application of simple rules to detect common errors like electrode detachment or high-amplitude bursts; (2) SVM classification using time-lagged covariance matrix elements as features; and (3) combination of scores from the first two stages to make the final decision. The simple rules provide a rapid and efficient means of identifying most noisy ECG segments, whereas the SVM classifier refines the classification by addressing borderline cases.

The method attained a score of 0.999 on the training dataset and 0.836 on the test dataset, demonstrating its effectiveness in filtering out low-quality ECG recordings and ensuring that only trustworthy data is sent to experts for further analysis. This approach not only saves valuable time for healthcare professionals but also improves the reliability of ECG monitoring using mobile devices.

Behar et al. (2013) [9] developed an automated algorithm to evaluate ECG signal quality during normal and arrhythmic conditions. The algorithm aimed to reduce false arrhythmia alarms in intensive care unit (ICU) monitors by extracting signal quality indices (SQIs) from ECG segments and employing an SVM classifier. The SQIs used in this study included kSQI, sSQI, pSQI, basSQI, bSQI, rSQI, and pcaSQI.

The proposed method was tested on three datasets: the PCCC 2011 (DB1), the MIT-BIH arrhythmia database (DB2), and the MIMIC II database (DB3). The ECG signals were manually annotated into two classes: good (A-B) and bad quality (D-E), while class C (borderline quality) was excluded to avoid confusion during model training. The distribution of signal quality across the databases was as follows: the ECG signals were manually examined and labeled by skilled cardiologists according to established clinical criteria. Only two primary categories were considered: optimal (A-B) and suboptimal (D-E). Segments that fell into the intermediate category (C) were intentionally excluded to minimize the likelihood of annotation discrepancies, thereby reducing the potential for confusion during the training process. The ecg signal were downsampled to 125Hz and QRS detection was performed using eplimited and wqrs algorithms.

The study achieved high classification accuracies, with accuracy rates reaching up to 99% for normal rhythms and 95% for arrhythmias. However, the performance varied across different types of rhythms. The study demonstrated that signal quality indices should be rhythm-specific, and classifiers should be trained for each rhythm independently to achieve optimal results, which requires a substantial amount of labeled data.

The performance of the SVM classifier is summarized in Table 3.8, which presents the results of training the classifier on datasets DB1-DB2 and testing it on dataset DB3:

Table 3.8: Performance Metrics for ECG Signal Quality Classification.

Arrhythmia	Bad	Good	Total	Ac	Se	Sp
AFIB	24	1261	1285	0.967	0.958	0.968
SVTA	0	12	12	1.000	0.000	1.000
AFL	3	81	84	0.833	1.000	0.827
SBR	1	333	334	0.961	1.000	0.961
VT	1	11	12	0.833	1.000	0.818
VFL	1	14	15	0.467	1.000	0.429
A	55	1453	1508	0.972	0.927	0.974
V	269	5603	5787	0.936	0.826	0.974
Overall	**269**	**8768**	**9037**	**0.946**	**0.863**	**0.948**

Abbreviations: AFIB - atrial fibrillation, SVTA - supraventricular tachyarrhythmia, AFL - atrial flutter, SBR - sinus bradycardia, VT - ventricular tachycardia, VFL - ventricular flutter, A - atrial premature beat, V - premature ventricular contraction.

3.2.2 Deep Learning Methods

Deep Learning is a subfield of ML that has revolutionized the field of AI, enabling computers to learn from large quantities of data and perform complex tasks with minimal human intervention. In contrast to traditional ML models, which often rely on manually engineered features, Deep Learning models learn automatic representations of the data, making them particularly useful for tasks that involve unstructured data such as images, audio, and biomedical signals [22].

The core of Deep Learning is the artificial neural network, particularly deep neural networks (DNNs), composed of multiple layers of nodes or **neurons**. These networks are able to model complex and non-linear relationships in the data by stacking multiple layers of transformation, each of which captures increasingly abstract features of the input data. The success of Deep Learning in various domains, from image recognition to natural language processing, is attributed to its ability to automatically extract relevant features from raw data, reducing the need for domain-specific expertise in feature engineering [36].

In the context of biomedical signal processing, particularly ECG analysis, Deep Learning has demonstrated extraordinary potential. Traditional methods for analyzing ECG signals often require extensive preprocessing and manual extraction of features to achieve accurate classification. Deep Learning models, on the other hand, can operate directly on raw ECG signals, learning to distinguish between normal and abnormal patterns without the need for extensive preprocessing. This capability has led to significant advances in automated diagnosis of various cardiac conditions, including arrhythmias and myocardial infarctions [26].

Furthermore, the development of Deep Learning architectures specialized for specific tasks, such as CNNs and recurrent neural networks (RNNs), has further improved the ability of these models to handle temporal and spatial dependencies inherent in ECG signals. CNNs are particularly effective in capturing local patterns in time-series data, while RNNs are well-suited for modeling sequential dependencies, making them ideal for analyzing the continuous nature of ECG signals [5].

Convolutional neural networks

CNNs are a specialized type of artificial neural network designed to process data with a grid-like topology, such as images. Unlike traditional neural networks, CNNs take advantage of the spatial structure of data, making them highly efficient and accurate for

tasks involving visual and spatial information. A typical CNN is composed of multiple layers that work together to extract and interpret features from raw data. These layers can be broadly categorized into three groups: convolutional layers, pooling layers, and fully connected layers and they are all represented in Figure 3.2.

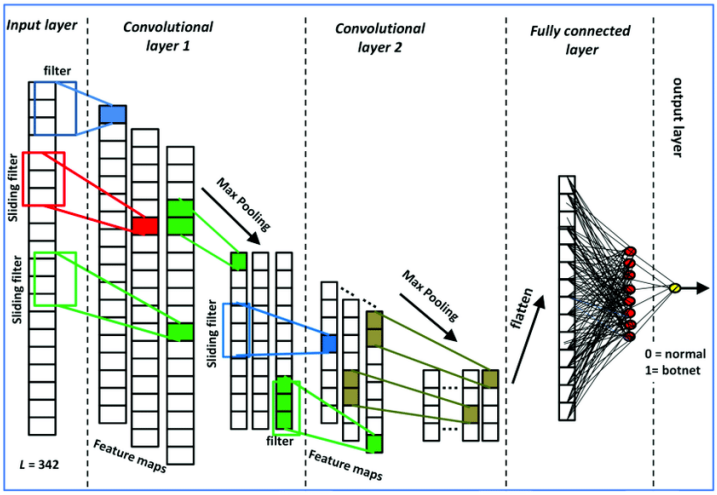


Figure 3.2: 1D-CNN model with two convolutional and max pooling layers feeding a dense fully connected layer [74].

The core component of a CNN is the **convolutional layer**, which relies on a filter or kernel to identify distinctive features within an input image. This process initiates by scanning the kernel across the image's width and height, covering the entire image multiple times. As a result, the input image is transformed into a set of feature maps or convolved features, each representing the presence and magnitude of a particular feature at various locations within the image. CNNs frequently incorporate numerous layers of convolutional processing, allowing the network to increasingly decipher the visual data present in the raw image. In the initial stages, the CNN identifies fundamental features such as boundaries, textures, or hues. As the layers progress, the CNN is able to extract more intricate patterns from the input data, capitalizing on the insights gleaned from the preceding layers.

The **pooling layer** complements the convolutional layer by condensing the input data, preserving essential information while simplifying the data set. This is accomplished through downsampling, which decreases the number of points in the input data, frequently achieved by reducing the number of pixels representing the image. Max pooling and average pooling are prevalent techniques used for downsampling.

The downsampling process significantly reduces the overall computational load and parameter count, resulting in improved efficiencies and generalization capabilities. By focusing on higher-level features, less complex models are less susceptible to overfitting, a common issue where the model becomes overly specialized to the training data, resulting in a significant decline in performance when faced with new, unseen information.

In the final stage of a CNN, the **fully connected layer** serves a critical function in image classification. This layer is distinguished by its intricate connections, where every neuron in one layer is connected to every neuron in the subsequent layer. By integrating the features extracted by convolutional and pooling layers, the fully connected layer enables the CNN to connect these features to specific outcomes or classes, as shown in Figure 3.2.

As a result, the CNN can take into account all features simultaneously when making a classification decision [17, 3].

Applications

Mondal et al. (2024) [44] designed a novel CNN for real-time and automatic ECG-SQA in wearable health monitoring devices with limited computational resources. The proposed CNN architecture consisted of a unique combination of convolutional layers and dense layers, optimized for efficient signal processing.

During the preparation of the ECG signal dataset, a series of pre-processing steps were implemented to refine data quality and optimize classification accuracy. Specifically, the ECG signals were divided into 5-second segments to exclude lengthy periods with brief noise disturbances. Subsequently, a two-stage filtering process was applied to each signal. Initially, a second-order high-pass Chebyshev filter with a cutoff frequency of 0.8 Hz was employed to eliminate baseline wandering artifacts. This was followed by a fourth-order low-pass Chebyshev filter with a cutoff frequency of 40 Hz to suppress high-frequency noise, including power line interference. To address amplitude variations between subjects, the ECG signals were magnitude-normalized before being presented to the CNN model. This rigorous pre-processing pipeline helped ensure that the CNN received high-quality input data for training and testing.

To achieve high performance, a thorough exploration of different activation functions and layer configurations was conducted. The selected architecture featured four convolutional layers, five dense layers, and utilized the exponential linear unit (ELU) activation function. The model was extensively trained and tested on a diverse set of ECG databases, including the PCCC 2011, PCCC 2017, and the St. Petersburg Institute of Cardiological Technics 12-lead arrhythmia database.

The study demonstrated excellent performance in classifying ECG signals as noisy or clean, with **sensitivity rates** of 92.88%, 82.09%, and 99.64%, and **specificity rates** of 75%, 75.3%, and 73.97% for the respective databases. Moreover, the model's efficiency was showcased through its compact size (5633 kB), rapid testing time (121.00 ± 39.77 ms), and low energy consumption (1851.3 ± 608.48 mJ) when implemented on a Raspberry Pi.

Zhang et al. (2019) [75] developed a cascaded CNN to tackle the complex issue of dynamic ECG-SQA, which is often compromised by artifacts and noise. In contrast to traditional binary classification methods, this study aimed to develop a more nuanced five-class classification system to cater to specific clinical needs. The dynamic ECG signals were categorized into three levels: low-interference, mild-interference, and severe-interference categories, each representing distinct types and severities of noise and artifacts.

The study employed a 12-lead Holter monitor to collect a dataset of 2,100 24-hour recordings from 2,100 subjects. The ECG signals were digitized at 128 samples per second and subsequently processed using a second-order Butterworth filter to remove high-frequency noise and baseline drift. The data were then divided into 4-second non-overlapping segments and labeled according to their corresponding interference types and severities.

The proposed cascaded CNN architecture consisted of two stages. The first stage utilized two subnetworks to distinguish between the types of signal interference (low-interference, myoelectrical noise, and motion artifacts). The second stage employed two

additional subnetworks to categorize the signals into mild or severe interference levels. This architecture leveraged both time-frequency spectrums and raw ECG signals for feature extraction, followed by classification using fully connected layers.

The trained model was validated on both a private dataset and the publicly available MIT-BIH Arrhythmia Database. The cascaded CNN achieved an overall recognition **accuracy** of 92.7% on the private dataset and 91.8% on the public dataset.

Huerta et al. (2020) [27] examined the performance of various CNN architectures for ECG quality assessment. The researchers employed transfer learning to evaluate the classification capabilities of five pre-trained CNN models, including AlexNet, VGG16, GoogLeNet, ResNet18, and InceptionV3.

The investigation utilized a subset of the PCCC 2017 database, comprising 5-second ECG intervals that were transformed into scalograms using Continuous Wavelet Transform (CWT). The primary objectives of the study were to assess the classification performance and computational efficiency of the various models.

The results indicated that all CNN models achieved high classification accuracy, ranging from 88.5% to 91.5%. Specifically, ResNet18 and InceptionV3 demonstrated the best performance, with **accuracies** of 91.10% and 91.25%, respectively. However, notable differences in computational load were observed. Despite its slightly lower accuracy, AlexNet emerged as the most efficient model in terms of CPU usage, memory requirements, and classification time (12.76 s).

Zhou et al. (2021) [79] focused on developing an automatic ECG quality assessment system that combines deep learning and conditional generative adversarial networks (CGANs) to enhance the efficiency of ECG signal filtering for clinical interpretation. The proposed system aims to reduce the workload for technicians and cardiologists by automating the process of identifying ECG signals of acceptable quality.

The system consists of two stages: a data enhancement stage and a quality evaluation stage. The first stage consists of a CGAN trained to generate segments of ECG signals, which are subsequently used to perform data augmentation of the training set in order to overcome dataset imbalance problems.

The quality evaluation model was initially trained using a combination of real and simulated ECG data and then fine-tuned using real ECG data. The model's performance was validated on three different datasets:

The dataset used in this research consisted of the following: the PCCC2017 Database, which was segmented into 10-second intervals, resulting in 555 unacceptable and 2618 acceptable ECG segments. The TELE ECG Database was combined with the PCCC2017 Database to form the COMD Dataset. Additionally, the Recreated Dataset (RECD) was created by introducing various types of noise (baseline wander, muscle artifacts, and electrode motion artifacts) into clean ECG recordings from the MIT-BIH arrhythmia database and the MIT-BIH Normal Sinus Rhythm database. The RECD dataset consisted of 7557 unacceptable and 20114 acceptable ECG segments.

The proposed system demonstrated high **accuracy** in ECG quality assessment, achieving 97.1% and 96.4% accuracy on the COMD and RECD datasets, respectively. In terms of **specificity**, the model achieved 96.4% on the COMD dataset and 95.0% on the RECD dataset. The **sensitivity** was equally impressive, with values of 98.6% for the COMD dataset and 99.1% for the RECD dataset. The study highlights the potential of CGANs in generating realistic ECG segments, which enhances the model's ability to generalize across different datasets.

Ma et al. (2023) [42] have developed innovative deep learning models for ECG-SQA in wearable devices, enhancing arrhythmia screening and diagnosis. The team proposed three models built upon the xResNet34 architecture, designed to handle noisy ECG signals collected during long-term wearable monitoring. The models were trained and evaluated using Brno University of Technology ECG Quality Database (BUT QDB) database, which categorizes signals into three classes: Good (all PQRST waves visible), Fair (only QRS complexes visible), and Poor (unsuitable for analysis).

Preprocessing involved normalizing signals to a range of 0 to 1 and downsampling to 200 Hz to reduce computational complexity. The models utilized signal quality indicators (SQIs) to assess the visibility of PQRST waves and QRS complexes, optimized using cross-entropy loss and stochastic gradient descent (SGD).

To verify the performance of these models, a 5 cross-validation technique was employed. The first model focused on arrhythmia screening, distinguishing between Good/Fair and Poor signals, achieving an average accuracy of 99.69%, sensitivity of 99.87%, and specificity of 98.83%. The second model targeted arrhythmia diagnosis, detecting Good signals with an accuracy of 96.40%, sensitivity of 97.15%, and specificity of 95.95%. The third model classified signals into three categories: Good, Fair, or Poor, reporting accuracies of 96.04% for overall classification, with Good signals achieving 96.62%, Fair signals 93.66%, and Poor signals 98.97%.

Huerta et al. (2021) [29] in 2021 emphasize the importance of ECG quality assessment in preventing misdiagnosis of cardiac disorders caused by noisy ECG signals, which are common in recordings obtained in real-world conditions.

In the study, the researchers compared the performance of a pre-trained CNN model, AlexNet, in classifying high- and low-quality ECG segments across two datasets. The first dataset consisted of 2,000 5-second ECG segments, equally divided between high-quality and low-quality signals, extracted from the PCCC 2017 database. The second dataset was generated through the application of data augmentation techniques, including time stretching, pitch shifting, noise addition, and amplitude modification, to the low-quality ECG segments. This process resulted in 1,000 additional low-quality intervals. To adapt AlexNet for this task, the ECG intervals were initially converted into 2D matrices utilizing a CWT. This technique converted the ECG signals into wavelet scalograms, which served as input for the CNN. By representing the ECG data as images, the CWT process enabled the use of AlexNet for this specific task, despite its primary design focus on 2D image data.

The researchers fine-tuned and tested the CNN model on both the original and augmented datasets and analyzed the results using a McNemar test. The statistical analysis showed no significant differences in classification accuracy, sensitivity, and specificity between the two datasets, suggesting that the synthesized noisy signals could be reliably used to train CNN-based ECG quality indices.

The study reported consistent performance metrics, with approximately 90% **accuracy**, **sensitivity**, and **specificity** across both datasets, indicating the effectiveness of the data augmentation approach used.

Tan et al. (2022) [62] developed a real-time quality assessment system for wearable multi-lead ECG data on mobile devices. To address the limitations of existing CNN models, which are often computationally expensive for mobile devices, the authors employed a resource-efficient neural architecture search algorithm, leveraging a modified version of ProxylessNAS, a neural architecture search algorithm capable of identifying efficient network architectures for subsequent deployment on mobile devices.

Prior to feeding the ECG data into the model, the researchers implemented a comprehensive preprocessing step to optimize signal quality and reliability. Initially, they utilized a band-pass filter with a frequency range of 0.5 Hz to 35 Hz to eliminate power frequency interference and baseline drift. This was followed by the application of a wavelet filter (Bior2.6) to further denoise the signals, thereby minimizing the impact of artifacts and ensuring that the input data was of high quality for the neural network.The system was evaluated on two datasets using 5-fold cross validation: a private wearable ECG dataset and the public PCCC 2011 dataset.

The experimental results demonstrated the system's excellent performance, with high accuracy and sensitivity across both datasets. Specifically, the system achieved an **AUC** of 98.32%, **sensitivity** of 85.68%, **specificity** of 97.02%, and an F1 score of 94.36% on the wearable dataset, and an **AUC** of 97.64%, **sensitivity** of 88.46%, **specificity** of 95.27%, and an **F1 score** of 93.52% on the PCCC dataset.

Moreover, the system demonstrated low latency, with an inference time of 78 ms on an Android emulator, making it suitable for practical use in mobile health applications.

Jin et al. (2023) [30] have introduced a novel deep learning model, DAC-LSTM (Dual Attentional Convolutional Long Short-Term Memory), aimed at enhancing the accuracy and interpretability of ECG quality assessment. This model addresses the challenge of reducing false alarms in automatic cardiovascular diagnoses and alleviating clinicians' workloads by assessing the quality of 12-lead ECG signals.

The DAC-LSTM model combines CNNs and bidirectional long short-term memory (BiLSTM) networks, along with a dual-layer **attention mechanism** that focuses on both channel-based and time-based attention. CNNs extract short-term features, while BiLSTM networks capture long-term dependencies. The attention mechanism enhances interpretability by highlighting the leads (channel-based) and time periods (time-based) that the model focuses on during the ECG quality assessment, making the model more clinically useful.

The study uses the PCCC 2011 dataset. The authors do not apply traditional preprocessing techniques such as noise filtering or signal denoising, opting to retain the original signal information, simplifying the process and minimizing time consumption while preserving the complexity of the ECG signals for the model to learn from directly.

The DAC-LSTM model outperforms existing methods, achieving a **sensitivity** of 97.59%, **specificity** of 76.47%, and **accuracy** of 94.0%, showing an average improvement of 3.35% in accuracy and 4.27% in sensitivity over other methods, with an inference time of 3.45 ms to predict each single ECG recording.

Below, in Table 3.9, are summarized the related works discussed previously, including preprocessing techniques, model architectures, and activation functions utilized. Additionally, the table highlights important performance metrics such as sensitivity, specificity, and accuracy, alongside the datasets used for evaluating each method.

Table 3.9: Summary of Deep Learning Related Works with Preprocessing and Model Details.

Ref	Preprocessing	Model	Activation Function	Layers	Se (%)	Sp (%)	Acc (%)	Dataset	CT (s)
[44]	DS, BP, AN, SG	1D-CNN	ReLU Leaky ReLU ELU	3 CL, 3 DL 3 CL, 3 DL 4 CL, 5 DL	89.09/99.26 81.75/95.09 92.88/99.64	66.48/61.34 57.69/48.50 75.00/73.97	86.28/80.30 78.76/71.80 90.66/86.80	PCCC2011/ In-house	-
[75]	BP, STFT, DS	1D-CNN, 2D-CNN	ReLU	NR	NR	NR	92.7 91.8	In-house MIT-BIH	-
[79]	DA (CGAN), BP, US, SG	1D-CNN, LSTM	Leaky ReLU ReLU Sigmoid	4CL, 1DL and 2LSTM	98.6 99.1	96.4 95.0	97.1 96.4	COMD RECD	-
[62]	BP, WF, SG	1D-CNN	ReLU	NR	97.02 95.27	85.68 88.46	94.55 93.50	In-house PCCC2011	78 ms x 12 NR
[30]	SG	DAC-LSTM	ReLU	5CL and 2 BiLSTM	97.6	76.4	94.0	PCCC2011	3.45 ms
[42]	AN, DS, SG	xResNet34	ReLU	3 CL 4 Stages with RB	99.87	98.83	99.69	BUT QDB	-
[27]	CWT	AlexNet VGG16 GoogLeNet ResNet18 InceptionV3	NR NR NR NR NR	5 CL 13 CL, 3DL 22 Layers 18 CL 3 CL	88.9 85.6 88.8 88.4 89.0	92.5 93.7 92.7 93.8 93.5	90.7 89.7 90.8 91.1 91.3	PCCC2017	12.76 ± 0.21 78.84 ± 0.57 25.33 ± 0.45 21.07 ± 0.36 76.86 ± 0.25
[29]	CWT, DA (TS, PS, NA, AM)	AlexNet	ReLU	5 CL	90.0	90.0	90.0	PCCC2017	-

Abbreviations: 1D-CNN - One dimensional Convolutional Neural Network, 2D-CNN - Two dimensional Convolutional Neural Network, DS - Downsampling, BP - Bandpass Filtering, CL - Convolution Layers, CT - Computational Time, DL - Deep Layers, ELU - Exponential Linear Unit, SG - Segmentation, AN - Amplitude Normalization, AM - Amplitude Modification, DA - Data Augmentation, CWT - Continuous Wavelet Transform, CGAN - Conditional Generative Adversarial Network, FC - Fully Connected, NA - Noise Addition, PS - Pitch Shifting, STFT - Short time Fourier Transform, TS - Time Shifting, US - Upsampling, WF - Wavelet Filtering, LeakyReLU - Leaky Rectified Linear Unit, ReLU - Rectified Linear Unit, RB - Residual Blocks.

3.2.3 Introduction to GANs

As observed in Table 3.9, many authors employed data augmentation techniques to address class imbalance, particularly by increasing the examples of the less represented class. Data augmentation is essential when working with datasets that have an uneven distribution of class labels, as it helps in creating more balanced datasets, ultimately leading to improved model performance. Traditional data augmentation techniques, such as stretching, scaling and noise addition have been commonly used to artificially expand datasets, but they often fail to introduce sufficient variability in the signals, offering the network examples that are only slightly different from the original data [14].

Current studies indicate, data augmentation with GAN has multiple advantages compared to traditional methods. GANs successfully learn intrinsic data distributions, therefore generating synthetic data that not only looks realistic but also diverse [14]. For example, [20] showed that the classification accuracy of medical images was better based on GAN-augmented than those from traditional augmented data. Because GANs are free to generate varied complex data, they seem to be a promising solution for studying class imbalance under difficult conditions, like in ECG signal classification.

Furthermore, it has been observed that GAN-based data augmentation not only increases the quantity of training and classification accuracy data but also enhances the diversity of the data, leading to more robust models that can generalize better to unseen examples [18]. This capability is particularly beneficial when dealing with borderline and rare classes, where traditional augmentation methods may fail to capture the subtle differences between classes.

GANs

GANs are an advanced type of ML model that consists of two distinct neural networks, often referred to as the **Generator** and the **Discriminator**. These networks work together through an adversarial process in which the Generator creates synthetic data, and

the Discriminator evaluates it in comparison to real data. The Generator's goal is to produce data that is nearly indistinguishable from actual data, while the Discriminator aims to accurately classify the generated data as synthetic [23]. GANs are classified into three primary elements:

- **Generative:** Refers to the process of generating data based on a learned distribution.
- **Adversarial:** Denotes the competitive nature of the training, with two networks working in opposition.
- **Networks:** Describes the deep learning framework used to build both the Generator and Discriminator.

The Generator typically operates as a CNN, generating new data instances by transforming random inputs into outputs that resemble the real data set. The Discriminator, functioning as a deconvolutional neural network, attempts to differentiate between the real data and the synthetic data created by the Generator.

This adversarial training forms a feedback loop: the Generator is penalized when the Discriminator successfully identifies its outputs as fake. Over time, as the training progresses, the Generator becomes increasingly adept at producing realistic outputs, and the Discriminator refines its ability to detect fake data. The ultimate goal is to reach a point where the Discriminator can no longer reliably distinguish between real and synthetic data [73].

The process of training GANs can be broken down into a few key steps:

1. The Generator produces an initial set of synthetic data.
2. The Discriminator evaluates both the synthetic data and real data, assigning probabilities based on the likelihood of each being real.
3. The Discriminator provides feedback to the Generator, which adjusts its approach to better mimic real data.
4. This cycle repeats until the synthetic data becomes sufficiently realistic, confusing the Discriminator.

GANs have proven to be useful in a variety of applications, from generating high-quality images to creating synthetic datasets for training other ML models. For instance, they have been used in image enhancement tasks such as super-resolution, where low-resolution images are upscaled to higher resolutions. GANs are also employed in areas such as data augmentation, where they generate new examples from existing data [58]. Below in Figure 3.3, the schematic representation of the functioning of a GAN is illustrated. The diagram shows the interaction between the Generator and Discriminator, where the Generator produces fake data and the Discriminator evaluates it in comparison to real data, creating an adversarial training loop.

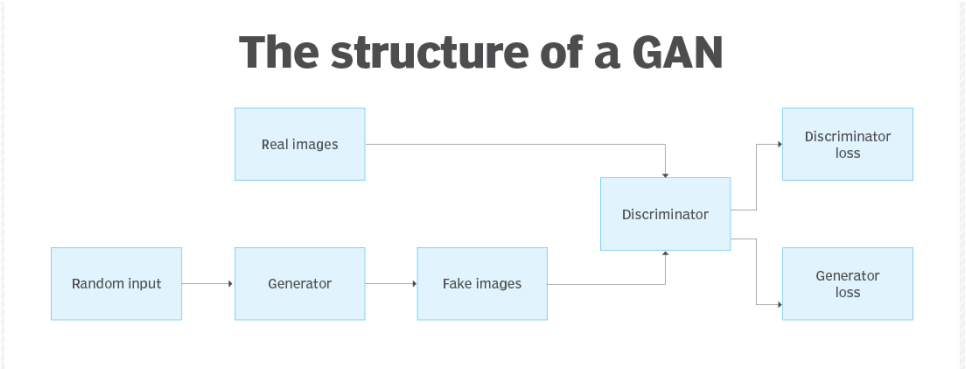


Figure 3.3: The structure of a GAN [73]

While GANs have demonstrated impressive capabilities in generating realistic synthetic data, they come with certain limitations.

One of the most prominent challenge with GANs is instability during training. Instability often leads to problems such as **mode collapse** [65] in which the generator starts to produce output with almost no variability instead of exploring the entire signal distribution. Another significant problem is the setting of hyperparameters for adversarial training process which if not done properly can lead to unreliable convergence. Furthermore, to date there are only a few quantitative indicators to analyze the quality of the results produced. Unfortunately, however, qualitative analysis still takes the lead [57].

These limitations can be summarized as:

- **Mode collapse:** The Generator converges to a state where it only produces a few types of outputs, failing to cover the full diversity of the data distribution.
- **Training instability:** Due to the adversarial nature of GANs, training can oscillate or diverge if not properly managed, which makes GANs difficult to optimize.
- **No clear evaluation metric:** outcome assessment is still done qualitatively. Although quantitative metrics have been developed, they have not yet completely replaced visual analysis.

In traditional GANs, the training is based on a min-max game between the **Generator** and **Discriminator**. The goal of the Generator is to minimize the probability of the Discriminator successfully identifying its outputs as fake, while the Discriminator maximizes its accuracy in differentiating real from fake data. The standard GAN loss function, called min-max loss [71], is based on the Jensen-Shannon (JS) divergence and it is given as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{3.7}$$

Where:

- G represents the Generator network,
- D represents the Discriminator network,
- $p_{\text{data}}(x)$ is the real data distribution,
- $p_z(z)$ is the noise distribution used as input to the Generator.

However, GANs using this approach often suffer from issues such as mode collapse and instability during training. To address these challenges, **WGAN** was introduced by Arjovsky et al. [7]

WGANs

WGAN replaces the JS divergence with the Wasserstein (**Earth Mover's**) distance, which provides smoother gradients and more stable training. The WGAN loss function is given as:

$$\min_{G} \max_{D \in \mathcal{D}_1} \mathbb{E}_{x \sim p_{\text{data}}(x)}[D(x)] - \mathbb{E}_{z \sim p_z(z)}[D(G(z))] \tag{3.8}$$

In this formulation, $\mathcal{D}_1$ represents the set of 1-Lipschitz functions, ensuring that the Discriminator remains constrained within this class of functions.

By imposing a strict limit on the magnitude of the Discriminator's weights, the **Weight Clipping** technique ensures that the model adheres to the 1-Lipschitz condition. This methodology has its own set of drawbacks, though. The implementation of Weight Clipping within WGAN is hindered by issues such as unstable training, low convergence when the clipping window is excessively broad, and the loss of gradients when the clipping window is too narrow [5].

WGAN with Gradient Penalty (WGAN-GP)

To address the limitations of weight clipping in WGAN, **Gulrajani et al.** [24] introduced **WGAN-GP**, which replaces weight clipping with a more effective method: **gradient penalty**. In WGAN-GP, the 1-Lipschitz constraint is enforced by penalizing the norm of the gradient of the Discriminator with respect to its input, ensuring that the gradient norm stays close to 1. The WGAN-GP loss function is given as:

$$L = \mathbb{E}_{\tilde{x} \sim p_{\text{data}}}[D(\tilde{x})] - \mathbb{E}_{z \sim p_z(z)}[D(G(z))] + \lambda \mathbb{E}_{\hat{x} \sim p_{\hat{x}}}\left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2\right] \tag{3.9}$$

Where:

- $\hat{x}$ represents samples drawn uniformly along straight lines between pairs of points from the real data and generated data distributions (p_{data} and p_g),
- λ is a penalty coefficient that controls the strength of the gradient penalty term.

The gradient penalty term ensures that the Discriminator's gradient norm remains close to 1, thereby satisfying the Lipschitz condition without resorting to weight clipping. As shown in Figure 3.4, it could be observed the issues that arise when using weight clipping in WGANs. The gradient norms either explode or vanish, which leads to instability during training. On the right, weight clipping forces the weights toward extreme values, resulting in poor gradient behavior. In contrast, the gradient penalty method maintains a more stable gradient norm and distributes the weights more evenly.

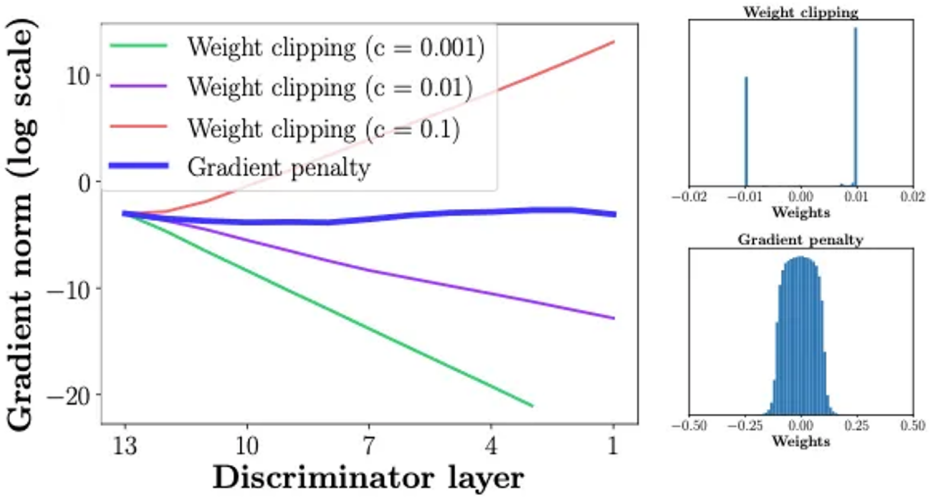


Figure 3.4: Comparison of Gradient Norms and Weights for Weight Clipping and Gradient Penalty. This shows the behavior of gradient norms across discriminator layers using weight clipping with different clipping values and gradient penalty. From [55].

3.3 Limitations of Prior Research and Study Rationale

The pursuit of advanced ECG signal quality evaluation has prompted a burst of innovative solutions, combining traditional and novel approaches. Although significant progress has been made, there remain gaps in understanding that warrant further exploration and creative problem-solving. Statistical methods based on SQIs provide a robust platform for automated ECG signal assessment, yet face challenges when applied to diverse datasets. These methods often require labor-intensive manual data processing, which can be time-consuming and may not fully capture the intricacies of the signal.

Notably, most of the works in the literature focus on the use of only one or at most two datasets, limiting their ability to generalize across diverse patient populations and varying ECG conditions. This narrow focus reduces the robustness of the models, particularly when applied to real-world data, which may differ substantially from the training datasets.

Moreover, very few studies have explicitly evaluated the applicability of these methods in real-life scenarios, where ECG signals are often noisy and irregular due to external factors, such as body movement or sensor placement in wearable devices. The models utilized in prior research, such as those in [29] and [42], tend to be computationally heavy. These models often require substantial computational resources and are not well-suited for battery-operated devices with limited computational power, such as smartwatches, holter monitors, and other wearable sensors.

Additionally, in many cases, k-fold cross-validation was used as a method to balance class distributions. While this technique can help mitigate class imbalance within the folds, it often results in over-representation of certain samples, leading to overfitting. As a consequence, the model may perform well on the training dataset but struggle to generalize in real-world deployments, where the data is more varied and complex, and class imbalances are more pronounced. Additionally, this approach fails to simulate real-world signal variations, thus overestimating the model's ability to handle unseen data in practical applications.

Another critical limitation of earlier work is the lack of focus on the classification of borderline signals. These signals, often falling between high-quality and poor-quality categories, are frequently misclassified or overlooked. The accurate identification of **borderline signals** plays a critical role in reducing false alarms and missed diagnoses, both of which are prevalent in real-time monitoring systems. This is particularly important in high-risk environments, such as intensive care units (ICUs), and in remote monitoring scenarios via wearable devices. By improving the detection and classification of these borderline signals, the study also enhances downstream tasks such as arrhythmia detection, heart rate variability analysis, and QT interval assessment, all of which are vital for the early detection of potential cardiac events.

To address these limitations, this study introduces several key innovations designed to enhance the robustness, efficiency, and real-world applicability of ECG signal classification systems:

- The integration of **multiple datasets** to ensure that the model can generalize effectively across diverse populations and a wide range of ECG signal conditions. This approach reduces the risk of overfitting to a specific dataset and enhances the model's adaptability to real-world scenarios, where data variability is much more pronounced.

- The development of a **dual-classifier system in a cascaded structure**, which refines decision-making for borderline signals that are often unregarded or misclassified by conventional methods. This cascaded approach allows for a more granular classification process, improving overall accuracy and reliability.

- The application of **Generative Adversarial Networks (GANs)** for data augmentation to overcome the issue of data scarcity. By generating synthetic yet realistic ECG signals, the dataset is balanced, improving the model's ability to learn from a more diverse set of examples and enhancing its performance in both training and testing phases.

- The implementation of a **real-time algorithm** specifically optimized for processing signals acquired from **wearable sensors**. This ensures that the system is compatible with resource-constrained devices such as smartwatches, holter monitors, and other wearable technologies, where real-time processing and energy efficiency are critical. The algorithm is designed to conserve battery life by applying computationally intensive preprocessing only to signals that require it.

- A thorough **analysis of computational complexity**, ensuring that the proposed method is both efficient and feasible for deployment in environments with limited computational power. This analysis guarantees that the system is not only accurate but also scalable and practical for real-time ECG monitoring applications, whether in clinical settings or through consumer health technologies.

The proposed methodology employs a **hybrid approach**, combining the strengths of deep learning for automatic feature extraction with machine learning classifiers for the precise categorization of borderline cases. This integration enables the system to maintain high performance even when confronted with noisy, complex, and real-world data, thereby improving its robustness and applicability across both clinical and consumer health applications.

Chapter 4

Materials and Methods

4.1 Dataset

In order to develop a reliable and effective classification model, it's crucial to work with multiple datasets. Each dataset offers a distinct set of characteristics, such as varying sampling rates, noise levels, and recording settings. By combining these datasets, we can gain a more detailed understanding of the model's ability to accurately classify ECG signals in real-world situations.

Developing a strong classification model needs a varied collection of ECG recordings. In the ensemble of datasets, not only high-quality signals were included but also also those with varying levels of noise and artifacts. Choosing a variety of datasets, for different scenarios, improves the capability of the model in classyfing ECG signals from real-life environment correctly.

By incorporating datasets with different signal qualities, such as high-quality signals and noisy signals with artifacts, the study aims to create a robust system capable of distinguishing acceptable from unacceptable ECG signals. Additionally, using datasets like the PhysioNet/Computing in Cardiology Challenge and the Brno University of Technology ECG Quality Dataset allows for comparison with previous studies, which helps validate the results of this study.

A key criterion for dataset selection was the focus on single-lead ECG signals, as the goal of this work is to develop a classification algorithm specifically designed for single-lead ECG monitoring. Single-lead ECGs are commonly used in wearable devices and mobile health applications, where simplicity and low power consumption are essential. By limiting the dataset to single-lead signals, the developed algorithm is intended to be practical for real-world use in such environments.

- **Brno University of Technology ECG Quality Dataset (QDB)**: QDB [47] was created by researchers at the Department of Biomedical Engineering, Brno University of Technology, in collaboration with the cardiology team. This dataset comprises long-term, single-lead ECG recordings and 3-axis accelerometer data collected from 15 participants, aged 21 to 83, over a period of 24 hours. The recordings were taken in the subjects' everyday environments, with the exception of swimming and bathing activities. The data was sampled at 1,000 Hz for ECG and 100 Hz for accelerometer data using a Bittium Faros 180 device.

  The dataset includes 18 ECG recordings, each with a minimum duration of 24 hours.

Three recordings are fully annotated for ECG quality, and the remaining 15 recordings have two 20-minute segments annotated for good signal quality, as well as five additional segments marked for poor signal quality. Signal quality is classified into four categories, with labelling performed on a sample-by-sample basis for each signal trace:

  - **Class 1**: High-quality signals where waveforms (P wave, T wave, and QRS complex) are clearly visible and reliably detectable.
  - **Class 2**: Signals with increased noise levels and unreliable detection of significant points. These signals are still readable but require additional processing.
  - **Class 3**: Signals with unreliable detection of QRS complexes, making them unsuitable for further analysis.
  - **Class 0**: Unlabelled samples where annotators did not provide a quality classification.

- **The PhysioNet/Computing in Cardiology Challenge 2017 (PCCC2017)**: The PCCC 2017 [15] focuses on the classification of atrial fibrillation (AF) from brief, single-lead ECG recordings.

  The challenge encouraged the development of algorithms capable of categorizing ECG signals into four classes: normal sinus rhythm, atrial fibrillation, other rhythms, and too noisy to classify.

  The dataset consists of 8,528 single-lead ECG recordings for training, with durations ranging from 9 seconds to over 1 minute. The data were collected using the AliveCor device, with a sampling rate of 300 Hz.

- **The MIT-BIH Noise Stress Test Dataset (NST)**: The NST Database [45] comprises 15 half-hour ECG recordings with 12 half-hour recordings of clean ECG signals from the MIT-BIH Arrhythmia Database, alongside three half-hour recordings of common noise types found in ambulatory ECGs.

  The noise recordings were generated using a combination of physical activity and standard ECG equipment, with electrodes positioned on limbs that do not capture the subjects' ECG signals. The noise types include baseline wander, muscle artifact, and electrode motion artifact.

  The noise records were then carefully blended with the clean ECG recordings from the MIT-BIH Arrhythmia Database (records 118 and 119) to create new ECG signals with varying SNRs. The noise levels were introduced in two-minute increments, alternating with two-minute clean segments.

  The resulting noisy ECG recordings span a range of SNRs from +24 dB to -6 dB.

- **TELE-ECG Dataset**: The TELE database [32] comprises 250 ECG lead-I signals collected from patients in a telehealth setting using the TeleMedCare Health Monitor. Following an initial set of 300 signals, 50 were discarded due to inconsistencies in artifact annotation, resulting in a final dataset of 250 usable records. Each ECG signal was digitized at 500 Hz using dry metal Ag/AgCl electrodes held in the patients' hands. Annotations for artifacts and QRS complexes were independently provided by three experts.

Table 4.1: Summary of Datasets Used in the Study

Dataset	Records	Duration	Leads	Sampling Freq. (Hz)	Acquisition
QDB	18	>24h	1	1000	Daily Environment
PCCC 2017	8528	30s - 1min	1	300	Laboratory
NST	15	30min	2	360	Hospital
TELE	250	Not Reported	1	500	Daily Environment

4.2 Pre-Processing

4.2.1 Summary of Preprocessing Steps

In this study, the various datasets come from different sources, each having distinct characteristics such as sampling frequencies, units of measurement, electrode placements, and recording environments. Therefore, a comprehensive preprocessing pipeline was essential to standardize these differences and ensure compatibility between the datasets.

The preprocessing steps included min-max scaling, downsampling, band-pass filtering, segmentation, and dataset-specific adjustments, such as annotator processing for QDB.

Min-max scaling normalized the signal amplitudes across all datasets, ensuring that values fell within the same range.

Downsampling reduced the computational load and unified the sampling frequency to 100 Hz across datasets, striking a balance between computational efficiency and the preservation of essential signal features.

Segmentation into 5-second windows was applied to allow the model to process uniform-length signal samples. Special care was taken to ensure that segmented windows from the same original trace were not mixed between training, validation, and test sets, preventing data leakage and ensuring that the model could generalize effectively. Finally, dataset-specific steps like annotator processing for the QDB dataset further ensured the integrity of labels and signal processing.

This preprocessing pipeline, as shown in Figure 4.1, aimed to make the signal data uniform in quality and format, allowing models to focus on meaningful features rather than discrepancies caused by variations in acquisition methods.

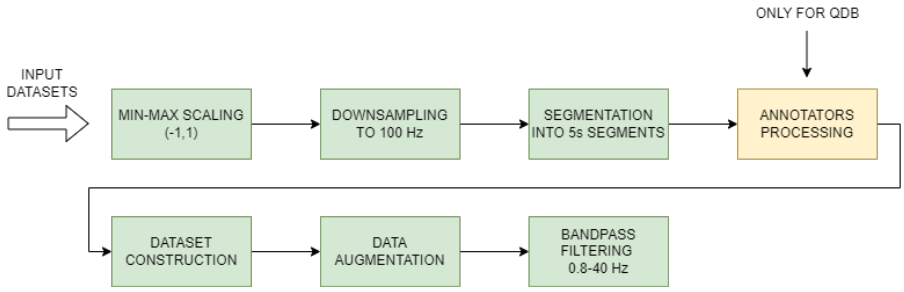


Figure 4.1: Overview of the ECG signal preprocessing steps. The green boxes represent steps common to all datasets, while the yellow box indicates a step exclusive to the QDB dataset.

4.2.2 Datasets handling

In the preprocessing phase, the first step was to handle datasets that were not initially combined. Each dataset was kept separate to ensure the integrity of the different sources

and their respective data characteristics. This approach was adopted to avoid mixing the signals, keeping track of individual signal traces and build a more reliable and robust model.

4.2.3 Scaling

As described in Subsection 4.2.2, the datasets were handled separately, and consequently, the normalization process was also applied individually to each dataset. Each dataset was processed one trace at a time, ensuring that the min-max scaling was performed independently on each dataset before subsequent preprocessing steps were applied. This approach preserved the integrity of the data sources and avoided introducing cross-dataset dependencies, ensuring uniform preprocessing within each dataset.

Min-max scaling was applied to normalize the dataset values within the range of -1 to 1. This normalization helps to standardize the input data, making it more suitable for ML models, particularly those that are sensitive to feature scaling. By scaling all the data uniformly, the model training becomes more stable and can converge more effectively. Additionally, min-max scaling helps to prevent large amplitude peaks in the ECG signals, such as QRS complexes, from disproportionately influencing the learning process.

This approach also ensures consistency across different datasets and recording devices, as ECG signals can vary significantly depending on the equipment used. Min-max scaling standardizes these variations, allowing the model to focus on the relevant signal features [54].

The formula for min-max scaling is as follows:

$$Y[n] = 2 \cdot \left(\frac{X[n] - \min(X)}{\max(X) - \min(X)} \right) - 1 \tag{4.1}$$

Where:

- $Y[n]$ is the scaled value of the signal at time step n,
- $X[n]$ is the original signal value at time step n,
- $\min(X)$ and $\max(X)$ are the minimum and maximum values of the original signal, respectively.

This formula rescales the input data such that all values lie within the range of -1 to 1, which is particularly useful, improving both model performance and training stability by making the data more homogeneous, reducing the likelihood of extreme values causing instability during the optimization process.

4.2.4 Downsampling

The original datasets were downsampled to 100 Hz to reduce computational load and align the datasets to a common frequency. However, this step introduces potential risks of subsampling errors, as downsampling can result in the loss of information, particularly in high-frequency components critical for ECG signal analysis. To evaluate the impact of subsampling, **Mean Squared Error (MSE)** and **Mean Absolute Error (MAE)** were calculated for downsampling at 100 Hz and compared with downsampling at 200 Hz. These metrics help quantify the deviation between the original and downsampled signals, providing insight into the accuracy of the resampling process.

Before the error calculation, a **Min-Max scaling** transformation was applied to each trace, ensuring that the signal values were scaled to a uniform range of [-1, 1]. This scaling step helps normalize the signal amplitudes across all traces, which is particularly useful when comparing the downsampled signals to their original versions.

For each trace, MSE and MAE were calculated by comparing the original signal with its downsampled and subsequently resampled counterpart (restored to the original frequency). The errors were then averaged across all traces to provide an overall assessment of the downsampling impact on the dataset.

The formulas for MSE and MAE are defined as follows:

Calculation of Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{4.2}$$

Where:

- n is the total number of data points in the signal.
- x_i is the original signal at sample i.
- $\hat{x}_i$ is the reconstructed signal at sample i.

Calculation of Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i| \tag{4.3}$$

Where:

- n is the total number of data points in the signal.
- x_i is the original signal at sample i.
- $\hat{x}_i$ is the reconstructed signal at sample i.

The errors were computed individually for each trace, and then averaged to provide the final MSE and MAE metrics shown in Table 4.2. This approach ensures that the variability across different traces is considered in the final error estimations.

Table 4.2: Downsampling Error Comparison (100Hz vs 200Hz) in Percentages

Dataset	Metric	100Hz (%)	200Hz (%)	Error Difference (%)
QDB	MSE	0.478%	0.116%	0.362%
	MAE	4.989%	2.591%	2.398%
PCCC2017	MSE	0.00118%	0.000205%	0.000975%
	MAE	0.138%	0.047%	0.091%
TELE-ECG	MSE	0.129%	0.0087%	0.1203%
	MAE	1.207%	0.127%	1.080%
NST	MSE	0.00968%	0.00316%	0.00652%
	MAE	0.486%	0.273%	0.213%

The analysis shows that downsampling to 100 Hz captures the peaks of interest in the ECG signals within the frequency range of 0.5-40Hz. This satisfies the Nyquist criterion

and avoids aliasing while maintaining computational efficiency. Given the small increase in error metrics and the significant reduction in data size, downsampling to 100 Hz is a practical choice for efficient analysis and processing. Further downsampling might lead to excessive loss of signal quality, hence, 100 Hz serves as an optimal balance between accuracy and computational cost.

4.2.5 Segmentation

Following the approach established in the literature, segmentation of the ECG signals into 5-second windows was applied. Specifically, Rahman et al. [50] evaluated different window durations and identified that a 5-second window provides the optimal balance between capturing relevant features in the ECG signals and ensuring efficient data processing. This segmentation was performed without overlap, with each 5-second window treated as an independent sample for subsequent processing steps.

In addition, the relevant labels were segmented along with the signals during segmentation so that each windows retain its correct annotation. Furthermore, the originating trace code for each window was carefully tracked across all datasets to guarantee that when the dataset is split in training, validation and test sets , windows came from the same initial trace will not be present in multiple datasets. This setup was necessary to avoid any form data leakage and to make sure that models do not see the same or very similar data during both training and evaluation. The above approach helps in creating a final model with less biased performance that generalizes well to unseen data.

4.2.6 Annotator Processing

The QDB dataset contains annotations for each sample of the ECG signal, provided by four independent annotators. Following the segmentation of the ECG signals and corresponding annotations into 5-second windows, an analysis of the annotations within each segment was conducted. For each 5-second window, the most frequent label among the four annotators was assigned as the final label for that segment through the **majority voting** method. This process ensures that the label assigned to each segment reflects the consensus among the annotators, thereby enhancing the dataset's reliability for subsequent model training.

During this process, any segments where the annotations consisted of only zeros (indicating that the annotators did not label these samples) were excluded from further analysis. This step was necessary to ensure that the dataset contained meaningful information. Only segments with at least one non-zero annotation were retained.

To evaluate the reliability and agreement among the four annotators, Fleiss' Kappa score was calculated. Fleiss' Kappa is a statistical measure used to assess the level of inter-rater agreement beyond chance and is particularly suitable for scenarios with more than two raters and categorical data [61]. The formula for Fleiss' Kappa (κ) is defined as:

$$\kappa = \frac{P - P_e}{1 - P_e} \tag{4.4}$$

Where:

- P represents the observed proportion of agreement among the raters.
- P_e represents the expected proportion of agreement due to chance alone.

Fleiss' Kappa ranges from -1 to 1, with the following interpretations:

- $\kappa = 1$ indicates perfect agreement among the annotators.
- $\kappa = 0$ indicates agreement equivalent to chance.
- $\kappa < 0$ indicates less agreement than expected by chance.

In this study, a Fleiss' Kappa score of $\kappa = 0.733$ was obtained, indicating a **good** strength of agreement according to the classification by Landis and Koch (1977) [35]:

- $\kappa < 0.20$: Poor agreement
- $0.21 \leq \kappa \leq 0.40$: Fair agreement
- $0.41 \leq \kappa \leq 0.60$: Moderate agreement
- $0.61 \leq \kappa \leq 0.80$: Good agreement
- $0.81 \leq \kappa \leq 1.00$: Very good agreement

The Fleiss' Kappa score of 0.733 reflects good agreement, supporting the reliability of the annotations used for training machine and deep learning models.

4.2.7 Dataset Construction

The construction of datasets for ECG signal classification is a crucial step in developing effective models. Both CNN and RaF models have distinct objectives, which require tailored datasets to ensure successful model training and evaluation.

As for CNN model, the binary classification requires to whether the given signal meets clinical quality standards or not. The dataset includes a diversity of signals both good for clinical analysis and not, to ensure overfitting is avoided and that generalization is robust on unseen data.

In contrast, the RaF model aims to classify signals into two categories: excellent quality and borderline quality. This approach helps streamline the signal processing workflow by identifying signals that require little to no further processing and those that may benefit from additional filtering or processing. The RaF dataset includes only signals that fall into these two quality categories, excluding non-acceptable signals from the analysis.

In summary, the CNN model focuses on distinguishing usable from non-usable signals, while the RaF model refines this classification further by identifying the level of processing required for usable signals. This structured approach to dataset construction ensures that both models can meet their specific objectives efficiently and effectively.

Dataset Preparation for CNN

For the CNN model, the dataset was constructed by combining multiple datasets, each prepared in accordance with specific criteria for acceptable and unacceptable signals.

For the **QDB dataset** signal belonging to Classes, 1 and 2 were considered as acceptable signals as opposed to Class 3 signals which were considered as unacceptable. Class 2 signals has been considered as acceptable because they could potentially be used for heartbeat detection or further filtered to extract relevant features.

For the **PCCC2017 dataset**, signals labeled as "normal sinus rhythm," "atrial fibrillation," and "other rhythms" were categorized as acceptable, while signals labeled as

"too noisy" were deemed unacceptable. Since this dataset includes traces with altered rhythms, the algorithm is being strengthened so that it focuses on signal quality rather than confusing pathological signals with signals that are qualitatively not good.

The **NST dataset** contains alternating segments of acceptable and corrupted signals (every 2 minutes). The first 5 minutes of clean signal were excluded from analysis, and to increase the number of unacceptable signals, segments with an SNR of -6 dB and 0 dB were used, constructed from clean signals from the MIT-BIH Arrhythmia dataset (traces 118 and 119). These were categorized as unacceptable.

Finally, in the **TELE-ECG dataset**, labels provided by Rahman et al. [50] were used. Only the poor-quality signals were included in the analysis, while the good-quality signals were excluded due to difficulties in categorization after visual inspection. The label distribution for acceptable and unacceptable signals in each dataset is shown in Table 4.3. For the CNN model, signals were classified as 0 (acceptable) or 1 (unacceptable) across all datasets.

Table 4.3: Distribution of Signals Across Datasets for Each Label (CNN)

Dataset	Acceptable Signals (Class 0)	Unacceptable Signals (Class 1)
PCCC2017	55000	1491
TELE-ECG	0	501
NST	0	1504
QDB	60691	10880

The datasets were split as follows: The PCCC2017 dataset was divided into training and validation sets with a 60%-40% split. The Tele-ECG dataset was used exclusively for training, while the NST dataset was used solely for validation. The QDB dataset was reserved for testing purposes. The division of the sets adhered to the criteria outlined earlier, ensuring that the data was split by subject, with careful attention to balancing the labels. This approach was taken to prevent any potential data leakage between the training, validation, and testing sets. Table 4.4 summarizes the final sets used for the CNN.

Set	Label 0	Label 1
Training	32990	1444
Validation	22010	2052
Test	60691	10880

Table 4.4: Final label distribution in Training, Validation, and Test sets.

Dataset Preparation for RaF

In the case of the RaF model, we utilize only labels 1 and 2 from the QDB dataset, which represent:

- **Label 1**: High-quality signals.
- **Label 2**: Borderline signals.

For the RaF model, these labels are reclassified as:

- **Label 1** is relabeled as **0**, representing high-quality signals.

- **Label 2** remains as **1**, representing borderline signals.

Signals labeled as 3, which correspond to unacceptable quality in the CNN model, are excluded from this analysis. Therefore, only high-quality (label 0) and borderline (label 1) signals are used in training and evaluation. The total number of high-quality and borderline signals across all subjects is summarized in Table 4.5.

Table 4.5: Total Distribution of High-Quality and Borderline Signals in QDB Dataset for Random Forest Model

Signal Type	Total Count	Percentage (%)
High-Quality Signals (0)	36,954	60.89%
Borderline Signals (1)	23,737	39.11%
Total	60,691	100%

Subsequently, signals from Table 4.5 were divided into two sets: training and validation, based on the trace codes. This division ensures that signals from the same trace do not appear in multiple sets, preventing data leakage and ensuring that the model's performance evaluation remains unbiased. It is important to note that the division was performed by keeping the entire traces intact, meaning that no trace was split between the training and validation sets. This approach further reduces the risk of data leakage and ensures the generalizability of the model's performance.

The following table summarizes the number of high-quality signals (labeled as 0) and borderline signals (labeled as 1) for each set:

Table 4.6: Distribution of High-Quality and Borderline Signals in Training and Validation Sets

Set	High-Quality Signals (0)	Borderline Signals (1)
Training	25267	17065
Validation	11687	6672

As shown in Table 4.6, the training set consists of 25267 high-quality signals and 17065 borderline signals. The validation set includes 11687 high-quality signals and 6672 borderline signals. These splits were designed to balance the number of signals in each class, ensuring a fair evaluation of the model's performance across different quality levels.

4.2.8 Data Augmentation

Data augmentation is a crucial technique employed to artificially increase the diversity and size of a dataset, particularly in the training of ML models like the CNN. It helps improve the model's generalization by creating variations of the existing data, such as adding noise, shifting, or stretching signals. This process is especially important when dealing with imbalanced datasets, as it reduces overfitting by exposing the model to a broader range of data points, ensuring it doesn't learn to rely on specific features from a limited set of examples [4].

In this context, two methods of data augmentation were used: traditional augmentation techniques and WGAN-GP (Wasserstein GAN with Gradient Penalty). Both methods incorporated signals of type White Gaussian Noise (WGN) with varying amplitude between 0.2 and 0.8. For the WGAN, 5000 synthetic signals were generated, while in the case of traditional data augmentation, only 2000 WGN signals were used. This was done to ensure that the augmented dataset for label 1 was not predominantly composed

of WGN examples. These techniques were applied exclusively to the underrepresented class (label 1) as part of the training set augmentation process, helping to balance the dataset and mitigate the issues associated with dataset imbalance and overfitting.

4.2.9 Traditional Data Augmentation

In this work, traditional data augmentation was performed by combining two main approaches to enhance the dataset's variability and size. Firstly, a 50% overlap was applied to the signals of each training dataset in the underrepresented class (label 1) to increase the amount of available data. Subsequently, to further enrich the dataset, each signal was subjected to various transformations with a 50% probability. These transformations included techniques such as time stretching, pitch shifting, noise addition, and amplitude modification, as highlighted in the study by **Huerta et al. (2021)** [29].

Additionally, as mentioned in Subsection 4.2.8 , 2000 white Gaussian noise (WGN) signals were added to the dataset. These WGN signals, however, were not subjected to any further transformations.

For the transformations applied, the following ranges of values were used:

- **Time Stretching:** The rate was selected from a uniform distribution between 0.2 and 2.0.
- **Pitch Shifting:** The number of pitch steps was randomly chosen between -3 and 3.
- **Noise Addition:** Noise was added with a signal-to-noise ratio (SNR) randomly selected between 0 and 2 dB.
- **Amplitude Modification:** The signal's amplitude was modified by a random factor within the range of -1 to 1 dB.

Figure 4.2 illustrates an example of a raw ECG signal and its corresponding transformations. The transformations show how time stretching, pitch shifting, noise addition, and amplitude modification introduce variability into the dataset, which is crucial for improving model generalization.

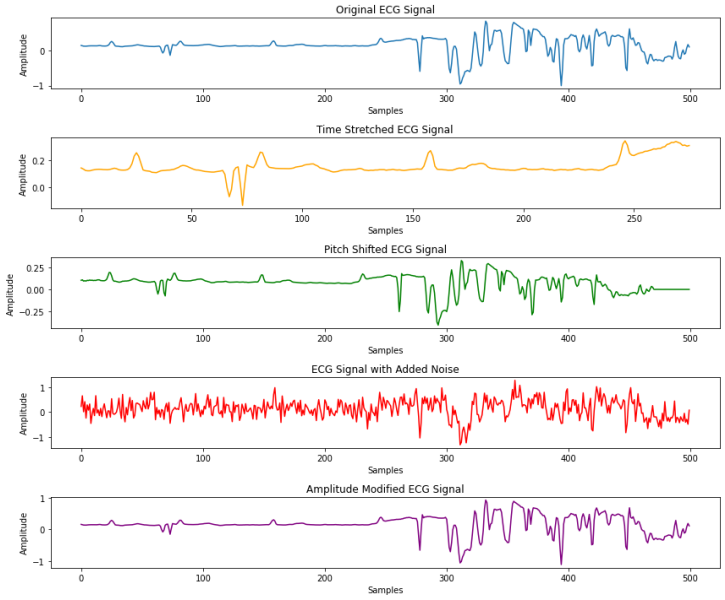


Figure 4.2: Example of transformations applied during data augmentation. The original ECG signal (top) undergoes time stretching, pitch shifting, noise addition, and amplitude modification.

The final dataset used for training , validation is represented in Table 4.7. This table shows the distribution of labels for the training, validation and test sets after applying data augmentation and preprocessing techniques.

Set	Label 0	Label 1
Training	32990	9136
Validation	22010	2052
Test	60691	10880

Table 4.7: Final label distribution in Training, Validation and Test sets.

4.2.10 WGAN-GP Data Augmentation

The implemented WGAN-GP (Wasserstein GAN with Gradient Penalty) model is composed of two primary neural networks: the generator and the discriminator (or critic). The architecture of both networks is designed to handle one-dimensional time-series data, specifically ECG signals of length 500. To match the input shape required by the model, the ECG signals were first reshaped by adding an extra dimension. The model architecture and training process were inspired by Keras WGAN-GP example [63]. The details of the networks are as follows:

Discriminator (Critic)

The discriminator, or critic, is built using multiple convolutional layers to extract meaningful features from the input ECG signals. The network is presented in Figure 4.3 and it includes the following components:

- **Input Layer:** The input to the discriminator is a signal of shape (500, 1), representing 5 seconds of ECG data sampled at 100 Hz.

- **Convolutional Layers:** Four convolutional layers are used with progressively increasing filters (64, 128, 256, 512) to capture various levels of abstraction in the ECG signals. Leaky ReLU activations are applied to introduce non-linearity, and dropout layers are added to prevent overfitting.
- **Output Layer:** The final output is a dense layer with a single neuron, which outputs a scalar value indicating whether the input signal is real or generated.

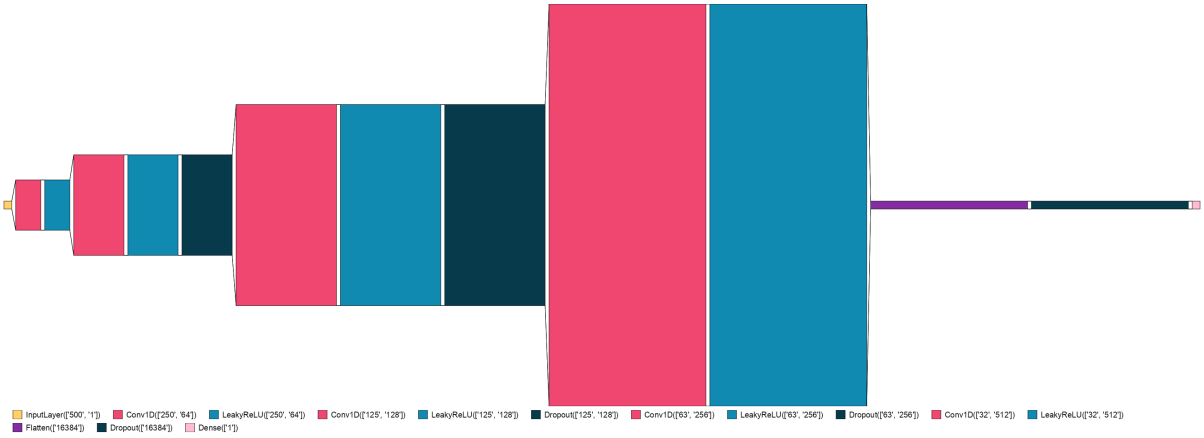


Figure 4.3: Critic Network Architecture.

Generator

The generator is responsible for transforming random noise vectors into realistic ECG signals. The architecture of the generator is presented in Figure 4.4 and it is composed as follows:

- **Input Layer:** The generator receives a random noise vector of size 128.
- **Dense Layer:** A fully connected dense layer reshapes the input noise into a suitable format for further upsampling. This dense layer is followed by batch normalization and a Leaky ReLU activation function.
- **Upsampling Layers:** The generator employs two upsampling layers with convolutional layers that progressively increase the length of the signal while refining the features. These layers ensure that the output has the desired shape and quality.
- **Output Layer:** The final output is passed through a convolutional layer with a `tanh` activation function, producing an ECG signal of length 500 and one channel. The `tanh` activation ensures that the generated values are normalized between -1 and 1, matching the scale of the real ECG signals.

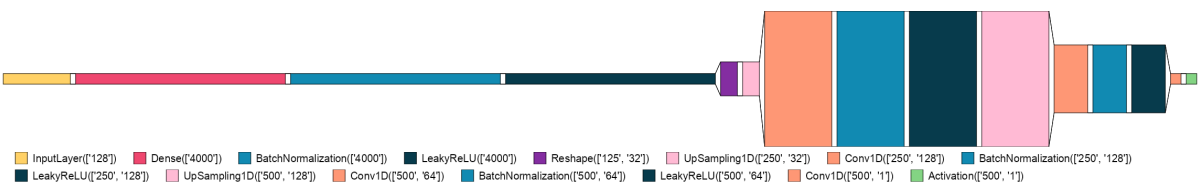


Figure 4.4: Generator Network Architecture.

4.2.11 Training Process

The training of the WGAN-GP was performed with the following setup and hyperparameters :

- **Input shape:** The input signals to the discriminator have a shape of 500 samples with 1 channel, corresponding to 5 seconds of ECG data sampled at 100 Hz.
- **Noise Dimension:** The generator takes random input noise vectors of size 128.
- **Batch Size:** The model was trained with a batch size of 512.
- **Optimizer:** The Adam optimizer was used for both the generator and discriminator, with different learning rates:
  - **Generator:** learning rate = 0.0002, $\beta_1 = 0.5$, $\beta_2 = 0.9$.
  - **Discriminator:** learning rate = 0.0003, $\beta_1 = 0.5$, $\beta_2 = 0.9$.
- **Loss Functions:**
  - **Discriminator Loss:** The loss was computed as the difference between the mean of the real data scores and the mean of the generated data scores.
  - **Generator Loss:** The generator's loss was the negative of the mean of the fake data scores, encouraging the generator to fool the discriminator.
- **Gradient Penalty:** A gradient penalty with a weight of 10.0 was applied to enforce the Lipschitz constraint on the discriminator, replacing the traditional weight clipping method.
- **Discriminator Extra Steps:** For every generator update, six discriminator updates were performed to enhance training stability.
- **Epochs:** The training was conducted for a total of 250 epochs, with the model and generated signals being saved every 30 epochs for monitoring purposes.

To monitor the progress, the model generated synthetic signals at the end of each epoch. These signals were saved as images for visual inspection. Additionally, the generator was saved at regular intervals during training to facilitate recovery and further analysis.

Figure 4.5 compares real ECG signals and synthetic signals generated by the WGAN model. The synthetic signals were generated after the training process and are compared to real ECG signals to assess the quality of the generated data.

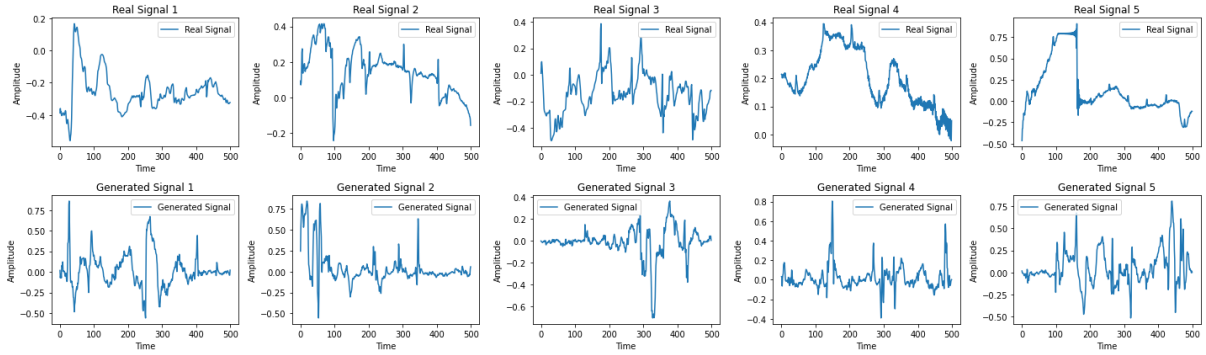


Figure 4.5: Comparison of Real and Generated ECG Signals. The top row shows the real ECG signals, while the bottom row shows the corresponding generated signals from the WGAN model.

In the end, 20,000 synthetic signals generated by WGAN-GP model were added to the training set for a more balanced dataset. In addition, as explained in Subsection 4.2.8, 5,000 WGN signals has been included in the training set. It resulted in 25000 new signals altogether which served the purpose of reducing this class imbalance and helped in generalization for both classes.

Table 4.8 summarizes the final label distribution in the Training, Validation, and Test sets, including the additional synthetic signals generated by the WGAN and WGN.

Set	Label 0	Label 1
Training	32990	26444
Validation	22010	2052
Test	60691	10880

Table 4.8: Final label distribution in Training, Validation, and Test sets. The Training set includes an additional 20,000 synthetic signals generated by the WGAN model and 5,000 WGN signals to balance the dataset.

4.2.12 Band-Pass Filtering

A band-pass filter was applied to the ECG signals, retaining frequencies between 0.8 Hz and 40 Hz. This filtering step was implemented for both CNN model and RaF model, prior to classification, following the approach used by Mondal et al. [44]. The filter employed was a Chebyshev Type I filter of order 5, chosen for its ripple characteristics in the passband, which provides stable and effective filtering. The primary goal of this filtering was to remove high-frequency noise, including network interference at 50-60 Hz, and baseline wanders, whose typical frequencies have been discussed in Subsection 2.1.2.

This range is also based on the established ECG frequency components, where the frequencies of interest for accurate peak detection, such as the QRS complex, typically do not exceed 50 Hz as discussed in Subsection 2.1.1 with a dominant frequency power from 10 to 40 Hz [66]. Thus, by filtering out unwanted noise and focusing on the crucial signal components, this step ensures that the algorithm receives clean and relevant data, improving the accuracy of feature extraction and classification. In Figure 4.6, the frequency response of the Chebyshev Type I bandpass filter used in this work is shown.

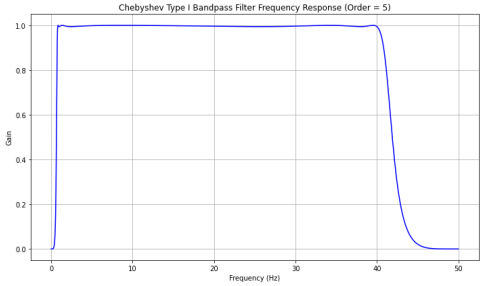


Figure 4.6: Frequency response of the Chebyshev Type I bandpass filter of order 5 used in preprocessing the ECG signals. The filter passes frequencies between 0.8 Hz and 40 Hz, removing high-frequency noise and low-frequency baseline wander.

4.3 Classification Algorithm

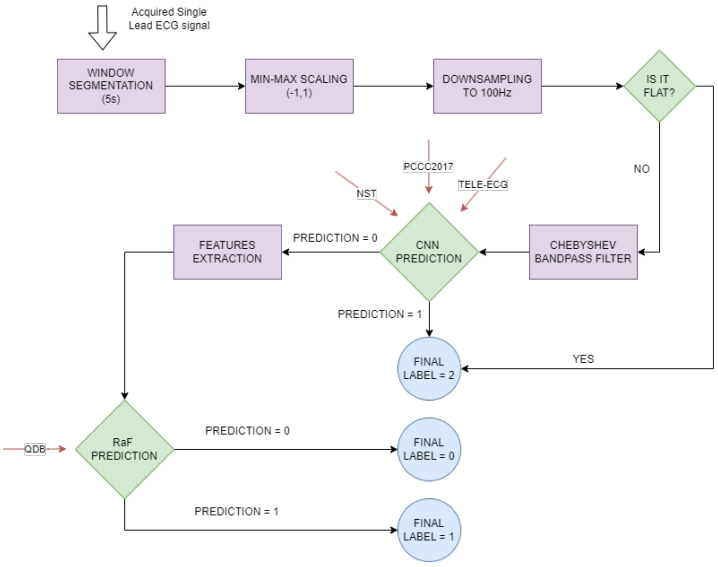


Figure 4.7: Flowchart depicting the classification algorithm. The ECG signals are preprocessed and passed through a CNN model and a RaF model in sequence, leading to a final classification into three labels (0, 1, 2). Each classifier is trained on different datasets, as shown.

As illustrated in Figure 4.7, the CNN model was trained with the PCCC2017 and TELE-ECG datasets, while NST was used for the validation set together with a PCCC2017 parcentage. It performs an initial binary classification. When the CNN predicts label 1, the signal is assigned label 2 directl

y. If the CNN predicts label 0, the signal is passed to the RaF model, which is trained with the QDB dataset and performs a secondary binary classification between label 0 and label 1. This cascade approach aims to create a final classification into three labels. Moreover, it should be noted that the CNN model did not learn QDB dataset features.

4.3.1 CNN

The CNN used in this work has been inspired by **Mondal et al.** [43] and it is presented in Figure 4.8. It is designed to process 1D ECG signals with a length of 500 samples. The

architecture consists of multiple 1D convolutional layers followed by pooling and dropout layers, which help prevent overfitting. LeakyReLU was used as the activation function after each convolutional layer to add non-linearity while mitigating the vanishing gradient problem.

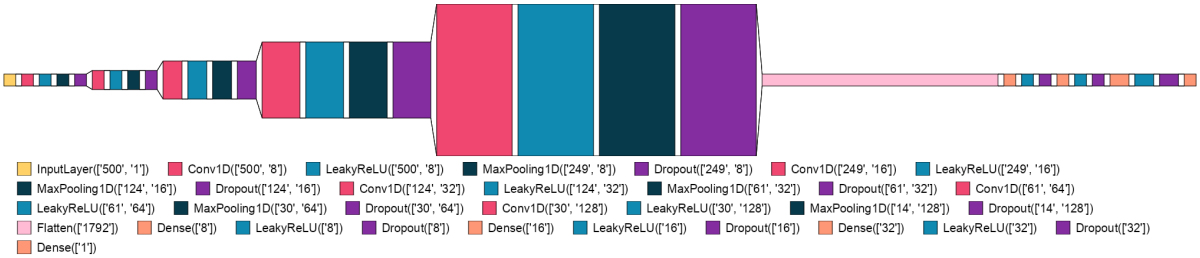


Figure 4.8: Architecture of the CNN model used for ECG signal classification. The network comprises five 1D convolutional layers with increasing filters, followed by pooling, dropout, and dense layers. LeakyReLU activations are applied after each convolutional and dense layer.

The architecture is summarized as follows:

- **Input:** The input layer receives ECG signals of shape (500, 1), where 500 represents the number of samples and 1 is the number of channels.
- **Convolutional Layers:** Five 1D convolutional layers with an increasing number of filters (8, 16, 32, 64, and 128) and a kernel size of 3. Each convolutional layer is followed by a LeakyReLU activation function with $\alpha = 0.2$ and max-pooling layers with a pool size of 3 and a stride of 2. Additionally, L2 regularization is applied in every convolutional layer to reduce overfitting.
- **Dropout:** After each pooling layer, a dropout layer with a dropout rate of 0.5 was applied to prevent overfitting.
- **Fully Connected Layers:** The output of the convolutional layers is flattened and passed through three fully connected dense layers with units of 8, 16, and 32, respectively. Each dense layer is followed by a LeakyReLU activation and dropout.
- **Output Layer:** The final output layer is a dense layer with a single unit and a sigmoid activation function, designed for binary classification tasks.

The total **number of parameters** in this model is 47,977, all of which are trainable. The model has no non-trainable parameters.

4.3.2 Random Forest (RaF) Classifier

In addition to the CNN, a RaF classifier was used in the cascade. RaF is a powerful ensemble learning technique that constructs multiple decision trees during training and outputs the mode of the classes for classification tasks. It is particularly useful in handling high-dimensional data and reducing the risk of overfitting.

For this work, the RaF classifier was implemented with the following parameters:

- **Number of Estimators:** 100 decision trees were trained in the ensemble, each independently learning a portion of the data.

- **Criterion:** Gini impurity was used to measure the quality of splits during tree construction. This criterion chooses the split that maximizes the homogeneity of the nodes.

The Random Forest model was chosen for its robustness, interpretability, and capability of handling diverse feature sets effectively. The ensemble of decision trees allows for more stable predictions and is less prone to overfitting compared to a single decision tree. Each decision tree in the RaF was trained on a bootstrap sample of the training data, and a random subset of features was considered for splitting at each node. This randomness helps create diverse trees that can generalize well to unseen data. The RaF classifier was trained on **features extracted** from the ECG signals:

- **Power Spectral Quality Index (PSQI):** Measures the proportion of signal power in the 5-15 Hz band relative to the broader 5-40 Hz band [50].
- **Signal-to-Noise Ratio (SNR):** Quantifies the ratio of the variance of the signal to the variance of noise [50].
- **Template Matching Quality Index (tmSQI):** The Template Matching Quality Index (tmSQI) was employed to evaluate the quality of the ECG signals based on their structural consistency. The tmSQI is calculated by first detecting the R-peaks within the ECG signal, which correspond to the prominent upward deflections representing the electrical depolarization of the heart's ventricles.

  Once the R-peaks are identified, individual heartbeats are extracted from the signal by creating a window around each R-peak. These windows typically capture the segment of the ECG surrounding the R-peak, encompassing the QRS complex, which contains crucial information about the heart's electrical activity.

  After extracting the individual heartbeats, a template is created by averaging the waveforms of these beats. This template serves as an ideal representation of a typical heartbeat for that specific signal. Each heartbeat is then compared to this template using a correlation coefficient, which measures the similarity between the individual heartbeat and the template.

  The tmSQI is derived as the average correlation across all beats within the signal. High tmSQI values indicate that the beats closely resemble the template, suggesting a consistent and high-quality signal, while lower values suggest increased variability and potential signal degradation.
- **Sample Entropy (SampEn):** SampEn is a commonly used measure to quantify the complexity and irregularity of physiological time-series signals, such as ECG signals. SampEn is particularly useful for identifying differences in signal quality, as it assesses the likelihood that similar patterns of data remain similar over increasing lengths.

  The formula for Sample Entropy [68] is given by:

$$H(x, m, r) = -\log\left(\frac{C(m+1, r)}{C(m, r)}\right)$$

  where:

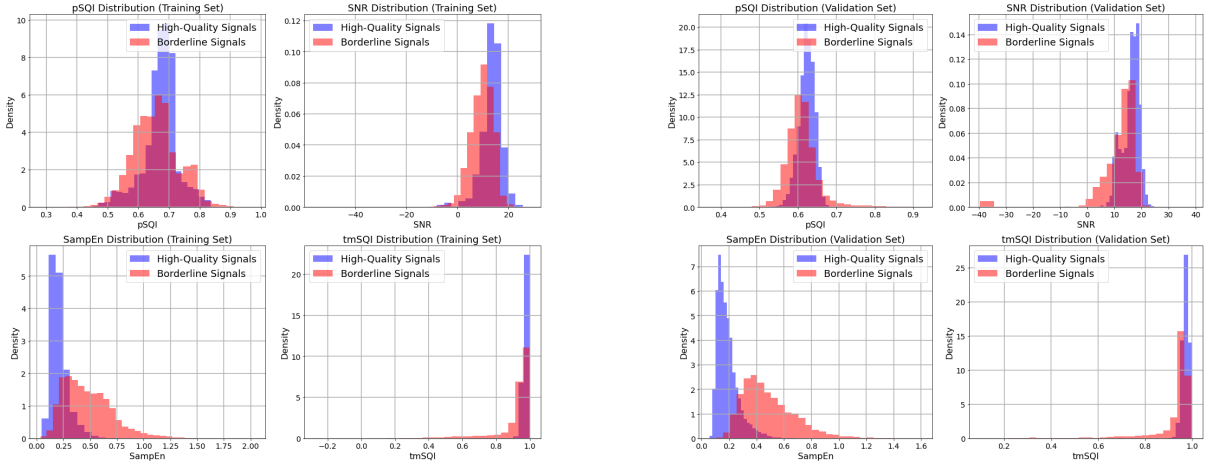
  – m is the embedding dimension (the length of the compared sequences),

- r is the radius of the neighborhood, typically set as $r = 0.2 \times \text{std}(x)$,
- $C(m+1, r)$ is the number of embedded vectors of length $m+1$ having a Chebyshev distance smaller than r,
- $C(m, r)$ is the number of embedded vectors of length m having a Chebyshev distance smaller than r.

Sample Entropy (SampEn) is an improvement over Approximate Entropy (ApEn), as it removes self-matches and therefore provides a more accurate and unbiased analysis of short data sets. Thus, SampEn is especially useful in clinical applications of assessing the complexity/regularity of ECG signals since clinical data are often with limited samples and lots of noise. In the present context, a lower SampEn value is representative of more regularity and self-similarity in the signal which are typically manifested by high-quality ECG signals. In contrast, an increased SampEn value usually indicates more randomness and disturbances, which is a quality of low-quality signals [53].

Figure 4.9 illustrates the distribution of key SQI across both the training and validation sets, comparing high-quality signals with borderline signals. The metrics included in the figure are as follows:

In both sets, high-quality signals (blue) exhibit distinct distributions compared to borderline signals (red), highlighting the variations in signal quality. These graphs demonstrate how different SQI characteristics can be used to distinguish between high-quality signals and signals that may have degradation or artifacts.



(a) Training Set (b) Validation Set

Figure 4.9: Comparison of SQI metrics for high-quality (blue) and borderline signals (red), for both training and validation sets.

4.3.3 CNN Training and validation

The CNN was tuned using a set of hyperparameters and callback mechanisms to enhance performance and mitigate overfitting:

- **Optimizer:** The *Adam optimizer* was used with an initial learning rate of 10^{-4}. A key benefit of using Adam over more traditional optimizers like *SGD* is its ability to converge faster, as it dynamically adapts the learning rate during training. This

allows for more efficient training, often leading to quicker convergence to optimal solutions [70].

- **Early Stopping:** The training process employed an early stopping mechanism that monitored the validation loss. If the validation loss did not improve for 23 consecutive epochs, training was terminated early, and the model weights were restored to those from the epoch with the **lowest validation loss**.

- **Learning Rate Adjustment:** In addition to the adaptive nature of Adam, the *ReduceLROnPlateau* callback was implemented. It reduced the learning rate by a factor of 0.1 when no improvement in validation loss was observed over 18 epochs. This dynamic adjustment facilitated convergence during training.

- **Model Checkpointing:** Model checkpointing was utilized to save the model with the best performance based on validation loss. This ensured that the most optimal version of the model was retained for further evaluation and testing.

- **Batch Size and Epochs:** The CNN was trained with a batch size of 512 and allowed to run for up to 100 epochs. However, the early stopping mechanism prevented excessive epochs by halting training once no further improvements in validation loss were detected.

- **Class Weight Balancing:** To address the imbalance between signal types, especially given the relatively scarce number of type 1 signals compared to type 0 signals, class weights were computed and applied during training. This adjustment ensured that the CNN model did not become biased towards the majority class, which is particularly important when traditional data augmentation techniques were used to balance the dataset. By setting class weights dynamically based on the distribution of classes, the training process was better equipped to handle imbalances, leading to a more robust and fair model. However, in the case of data augmentation using WGAN, the need for class weight balancing had a reduced impact because the dataset was already more balanced due to the effective generation of synthetic data. This balancing allowed the model to focus more on learning the intricate features of the signals rather than compensating for class distribution disparities.

These hyperparameters were selected based on a tuning and validation process aimed at maximizing generalization performance on unseen data. The analyses were performed on a machine equipped with an Intel64 Family 6 Model 170 Stepping 4, GenuineIntel CPU, which features 16 cores and 22 threads. The system also had 31.4 GB of RAM and was supported by an NVIDIA GeForce RTX 4070 Laptop GPU with 8188.0 MB of memory. .

4.3.4 Performance Evaluation

The performance of both the CNN and RaF models is evaluated using various metrics, including Accuracy, Recall, Specificity, Precision, F-Score, AUC, and ROC curves. For the CNN model, the loss curves and their moving averages for both the training and validation sets are shown below in Figure 4.10.

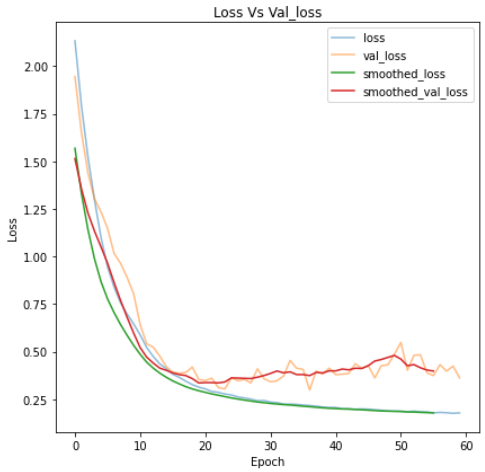


Figure 4.10: Training Loss vs Validation Loss during training with smoothed moving averages. The smoothed curves help visualize the trend and stability of loss over epochs.

The evaluation metrics used in this analysis are defined as follows. *Note that in this case, is being referred to positive instances as good signals (labeled as 0) and negative instances as bad signals (labeled as 1)*:

- **Accuracy** is the ratio of correctly predicted observations to the total observations. It is calculated using the formula:

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

  where TP and TN are true positives and true negatives, respectively, and FP and FN are false positives and false negatives.

- **Recall (Sensitivity or True Positive Rate)** measures the ability of the model to correctly identify positive instances. The formula is:

$$\text{Recall} = \frac{TP}{TP + FN}$$

  where TP represents the true positives and FN the false negatives.

- **Specificity (True Negative Rate)** evaluates how well the model identifies negative cases:

$$\text{Specificity} = \frac{TN}{TN + FP}$$

  where TN represents the true negatives and FP the false positives.

- **Precision (Positive Predictive Value)** quantifies how many predicted positives are actually correct:

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

  where TP represents the true positives and FP the false positives.

- **F-Score** is the harmonic mean of Precision and Recall, providing a balance between the two:

$$\text{F-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Area Under the Curve (AUC)** is a single scalar value that represents the model's ability to distinguish between positive and negative classes across different decision thresholds. The ROC curve plots the true positive rate against the false positive rate.

4.3.5 Algorithm Complexity Analysis

The computational time required for each phase of the algorithm was calculated to ensure that the processing times are suitable for real-time implementation. Specifically, the computational time was computed separately for the preprocessing phase, which includes signal segmentation and band-pass filtering, the CNN prediction phase, where the deep learning model processes the signal to classify it, and the RaF prediction phase, which performs further classification based on the extracted features.

In addition to the time measurements, the computational complexity of the CNN model has been assessed by calculating the FLOPS (Floating Point Operations Per Second) and MACs (Multiply-Accumulate Operations).

FLOPS represent the total number of floating-point operations required by the CNN during inference, which gives an estimate of the computational workload needed to process each input. MACs, on the other hand, quantify the number of multiply-accumulate operations, which are essential in convolutional layers, where input features are convolved with learned filters to produce the output feature maps [37].

Chapter 5

Results

5.1 CNN Results

The performance of the CNN model was evaluated on the training, validation, and test sets using the two different data augmentation techniques discussed in Subsection 4.2.10 and Subsection 4.2.9. It is also recalled that the datasets were divided as previously discussed in Subsection 4.2.7: The PCCC2017 dataset was split into training and validation sets with a 60%-40% split. The Tele dataset was used exclusively for training, while the NST dataset was used solely for validation. The QDB dataset was reserved for testing purposes. Below are the ROC curves for each dataset and data augmentation strategy, followed by a summary table of evaluation metrics.

5.1.1 Traditional Data Augmentation Results

The ROC curves for the training, validation, and test sets with traditional data augmentation are shown in Figure 5.1.As observed, the training and validation ROC curves are almost completely overlapping, indicating that the model performs very similarly on both sets. However, the test set curve shows a noticeable difference, highlighting that the model struggles with generalization when evaluated on unseen data.

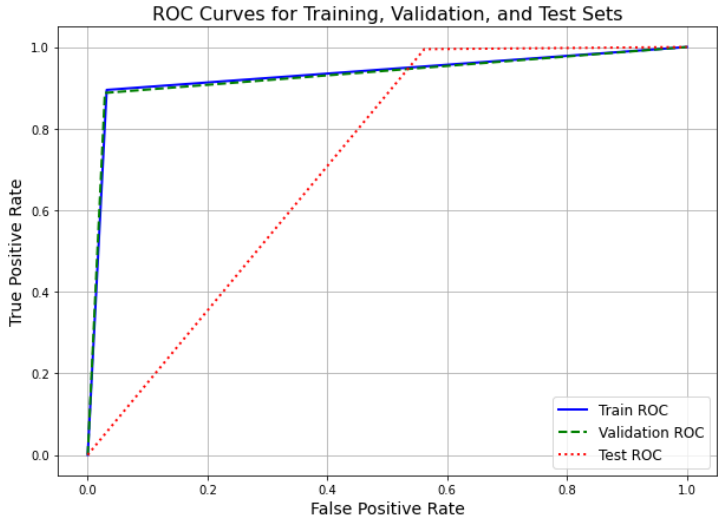


Figure 5.1: ROC Curves for Training, Validation, and Test Sets with Traditional Data Augmentation

This is reflected in the performance metrics outlined in Table 5.1, where the test metrics significantly drops compared to the validation set.

Metric	Training (Traditional)	Validation (Traditional)	Test (Traditional)
Accuracy	0.9528	0.9642	0.5229
F1 Score	0.8917	0.8086	0.3879
AUC Score	0.9864	0.9806	0.9855
Recall	0.8965	0.8874	0.9947
Precision	0.8869	0.7427	0.2410
Specificity	0.9684	0.9713	0.4383

Table 5.1: Evaluation Metrics for CNN Model with Traditional Data Augmentation

5.1.2 GAN-Based Data Augmentation Results

The ROC curves for the training, validation, and test sets with GAN-based data augmentation are presented in Figure 5.2. Compared to the traditional data augmentation, the model shows better generalization on the test set, with the ROC curves being more closely aligned between the training, validation, and test sets.

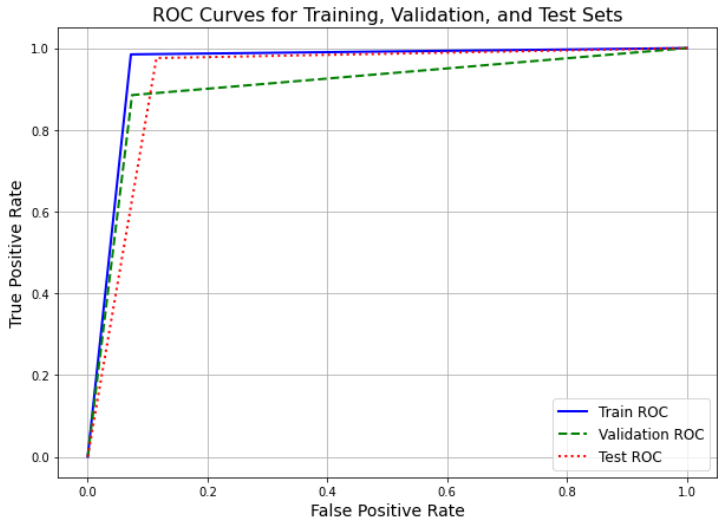


Figure 5.2: ROC Curves for Training, Validation, and Test Sets with GAN-Based Data Augmentation

This is further corroborated by the higher accuracy, F1 score, and AUC scores in Table 5.2, suggesting that GAN-based augmentation was more effective at improving model generalization.

Metric	Training (GAN-Based)	Validation (GAN-Based)	Test (GAN-Based)
Accuracy	0.9532	0.9225	0.8991
F1 Score	0.9493	0.6608	0.7461
AUC Score	0.9902	0.9666	0.9650
Recall	0.9846	0.8850	0.9756
Precision	0.9163	0.5273	0.6040
Specificity	0.9279	0.9260	0.8853

Table 5.2: Evaluation Metrics for CNN Model with GAN-Based Data Augmentation

From the results, it is evident that the traditional data augmentation approach led to poor generalization, as reflected in the steep decline in test accuracy and F1 scores compared to the validation set. This suggests that the model overfitted the training data and struggled to adapt to new, unseen data. On the other hand, GAN-based augmentation provided more balanced results, with comparable metrics across the training, validation, and test sets. This indicates that GAN-based data augmentation was more effective in mitigating overfitting and enhancing the model's generalization ability. While the test set performance with GAN-based augmentation was not perfect, it remained significantly better than the traditional approach, as indicated by the higher test accuracy and F1 scores.

5.2 RaF Results

The performance of the RaF model was evaluated using the QDB dataset, divided into training and validation sets, as described in Subsection 4.2.7. This division ensures that the model was trained and validated on separate portions of the data, thereby preventing overfitting and enabling an accurate evaluation of its generalization performance.

Figure 5.3 shows the confusion matrix of the training set, demonstrating the RaF model's near-perfect classification performance, with only minimal misclassifications. The model achieves an almost perfect recall and precision, as indicated by the minimal number of incorrect predictions.

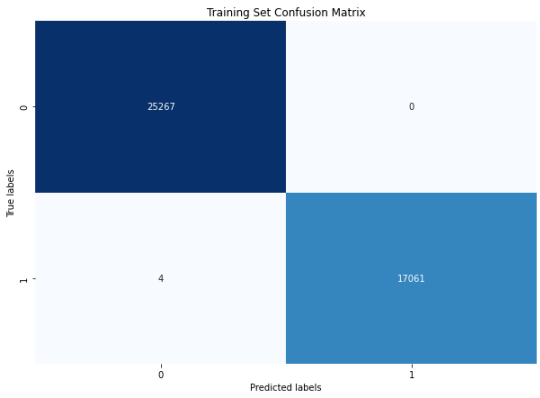


Figure 5.3: Training Set Confusion Matrix for Random Forest

In contrast, Figure 5.4 presents the confusion matrix for the validation set. Although the validation set performance is lower compared to the training set, the RaF model still manages to correctly classify the majority of the samples, achieving an accuracy of 85.83%. The confusion matrix shows a larger number of misclassified samples, especially in Class 1.

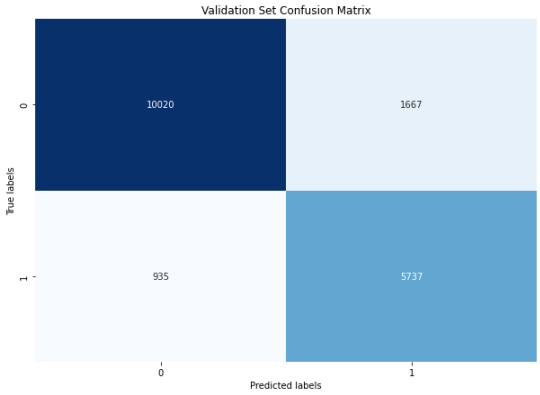


Figure 5.4: Validation Set Confusion Matrix for Random Forest

As shown in Table 5.3, the RaF model demonstrates strong performance on both the training and validation sets. While the training set achieved near-perfect results with an accuracy and F1 score of 99.99%, the model also performed robustly on the validation set, with an accuracy of 85.83%. This indicates that the model generalizes well to unseen data. The recall of 85.99% and the AUC score of 93.73% on the validation set further underscore the model's effectiveness in distinguishing between high-quality and borderline signals, showcasing its capability to maintain reliable classification across different data sets.

Metric	**Training Set**	**Validation Set**
Accuracy	0.9999	0.8583
F1 Score	0.9999	0.8151
AUC Score	1.0000	0.9373
Recall	0.9998	0.8599
Precision	1.0000	0.7749
Specificity	1.0000	0.8573

Table 5.3: RaF Model Evaluation Metrics on Training and Validation Sets

5.3 Final Cascade Model Results

The final cascade model, which integrates the CNN and RaF classifiers, demonstrates a balanced performance across all classes. The confusion matrix in Figure 5.5 visualizes the classification results on the validation set, highlighting how well the cascade model distinguishes between high-quality (Class 0), borderline (Class 1), and unacceptable (Class 2) signals. This validation set was specifically chosen as it was not used during training for either the CNN or RaF models, ensuring unbiased evaluation.

From the confusion matrix, it can be observed that the cascade model achieved strong classification accuracy, especially for Class 2 (unacceptable signals), where almost no misclassifications occurred. There was some overlap between Class 0 (high-quality) and Class 1 (borderline) signals, as evidenced by the misclassification of some Class 0 signals as Class 1 and vice versa.

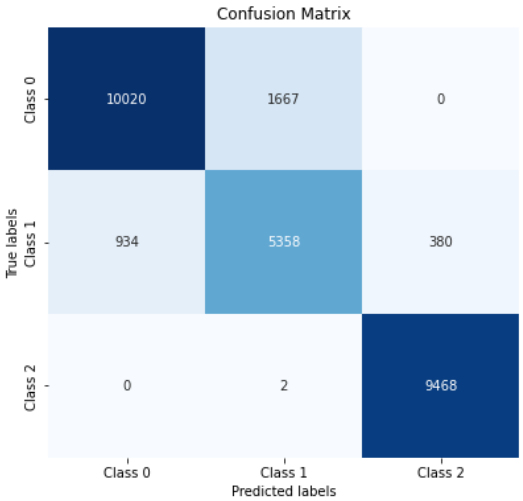


Figure 5.5: Confusion Matrix of the Final Cascade Model on the Validation Set

This overlap is further reflected in the performance metrics in Table 5.4. While the precision and recall scores for Class 2 were near perfect, the performance for Class 1 was slightly lower.

As summarized in Table 5.4, the overall accuracy of the cascade model stands at 89.28%. Despite the drop in precision and recall for Class 1, the model still performs well across all categories, making it a reliable system for differentiating between signal qualities. The model's high recall and specificity values for Class 2 further emphasize its robustness in identifying unacceptable signals, a critical requirement for real-time monitoring systems.

Class	Precision	Recall	F1-Score	AUC	Specificity
0	0.9147	0.8574	0.8851	0.9740	0.9421
1	0.7625	0.8031	0.7822	0.9228	0.9211
2	0.9614	0.9998	0.9802	0.9895	0.9793
Overall Accuracy					**0.8928**

Table 5.4: Class-wise Metrics and Overall Accuracy

5.4 Algorithm Complexity Results

The computational time required for each phase of the algorithm has been measured and reported in Table 5.5:

Phase	Time (seconds)
Preprocessing	0.0010
CNN Prediction	0.2292
RF Prediction	0.0156
Total Time	**0.2458**

Table 5.5: Computational time for each phase of the data processing pipeline.

As shown in Table 5.5, the overall processing time for a single 5-second ECG signal amounts to approximately 0.2458 seconds, which is relatively small. The most time-consuming step is the CNN prediction, which takes about 0.2292 seconds.

The preprocessing phase, which includes signal segmentation, downsampling, normalization through min-max scaling, and filtering using a Chebyshev band-pass filter, takes only 1 ms. This phase is illustrated in Figure 4.7, where the entire algorithm pipeline is shown in detail.

In addition to computational time, the complexity of the CNN model has been analyzed in terms of FLOPS and MACs. For the implemented CNN, the results are as follows:

- **MACs:** 1.42 MMACs
- **FLOPS:** 2.89 MFLOPS

These values suggest that the implemented CNN is computationally efficient and suitable for real-time applications. When compared to other 1-dimensional CNN architectures used in lightweight signal processing tasks, the obtained FLOPS and MACs are relatively low. For instance, the 1D-CNN architecture described in [62], which was designed for mobile devices, required approximately 36.44 MFLOPs, indicating that this CNN has an even lower computational load.

This comparison highlights that the CNN model is well-optimized, effectively balancing processing speed and computational efficiency, which is essential for maintaining real-time performance without compromising accuracy. The fact that the architecture in [62] was intended for mobile devices further underscores the efficiency of the CNN developed in this work.

Chapter 6

Discussion

6.1 CNN Performance

The CNN model exhibited high performance on both the training and validation sets, achieving accuracy rates of 95.32% and 92.25%, respectively, when employing GAN-based data augmentation. The model's performance on the test set demonstrated robustness, with an accuracy of 89.91%. This is a key strength of the model, as the performance on the test set closely matches that of the validation set, despite the differences in traces and datasets used. This similarity in performance across diverse datasets highlights the model's generalization capability, ensuring consistent results even when exposed to new, unseen data.

Compared to traditional data augmentation methods, which produced noticeably lower test accuracy of 52.29%, this suggests that the GAN-based augmentation improved generalization.The enhancement of performance metrics, like F1 score and AUC score, provides additional evidence of the efficacy of GAN-based augmentation in producing more varied and representative datasets for training models.

However, it is important to note that the Precision and F1 Score are relatively lower compared to other metrics, particularly on the test set. This discrepancy can be largely attributed to the imbalance present in the test dataset. In highly imbalanced datasets, where one class significantly outnumbers the other, metrics like Precision and F1 Score can suffer. These metrics are highly sensitive to the correct classification of the minority class (in this case, the "1" labels representing unacceptable signals).

Since Precision depends on the number of True Positives (TP) relative to the sum of TP and False Positives (FP), and F1 Score is the harmonic mean of Precision and Recall, any misclassification of the majority class as the minority class (e.g., classifying "0" signals as "1") can significantly affect these metrics. Given the imbalance, even a small number of incorrect classifications (i.e., signals that were actually "0" but were classified as "1") can disproportionately impact Precision and F1 Score. As a result, while the model may perform well overall, the Precision and F1 Score appear lower because the test dataset contains far fewer "1" labels, making these metrics more sensitive to errors in this class.

6.1.1 Comparison of Deep Learning-Based ECG Classification Methods

This section provides a comparative analysis of various studies that utilized Deep Learning models for ECG signal quality classification, focusing on their respective architectures,

datasets, and key performance metrics. The studies are summarized in Table 6.1, which highlights differences in sensitivity (Se), specificity (Sp), accuracy (Acc), and computational time (CT).

Table 6.1: Summary of Deep Learning Related Works with Preprocessing and Model Details.

Ref	Preprocessing	Model	Activation Function	Layers	Se (%)	Sp (%)	Acc (%)	Dataset	CT (s)
[44]	DS, BP, AN, SG	1D-CNN	ReLU Leaky ReLU ELU	3 CL, 3 DL 3 CL, 3 DL 4 CL, 5 DL	89.09/99.26 81.75/95.09 92.88/99.64	66.48/61.34 57.69/48.50 75.00/73.97	86.28/80.30 78.76/71.80 90.66/86.80	PCCC2011/ In-house	-
[75]	BP, STFT, DS	1D-CNN, 2D-CNN	ReLU	NR	NR	NR	92.7 91.8	In-house MIT-BIH	-
[79]	DA (CGAN), BP, US, SG	1D-CNN, LSTM	Leaky ReLU ReLU Sigmoid	4 CL, 1 DL and 2 LSTM	98.6 99.1	96.4 95.0	97.1 96.4	COMD RECD	-
[62]	BP, WF, SG	1D-CNN	ReLU	NR	97.02 95.27	85.68 88.46	94.55 93.50	In-house PCCC2011	78 ms x 12 NR
[30]	SG	DAC-LSTM	ReLU	5 CL and 2 BiLSTM	97.6	76.4	94.0	PCCC2011	3.45 ms
[42]	AN, DS, SG	xResNet34	ReLU	3 CL 4 Stages with RB	99.87	98.83	99.69	BUT QDB	-
[27]	CWT	AlexNet VGG16 GoogLeNet ResNet18 InceptionV3	NR NR NR NR NR	5 CL 13 CL, 3DL 22 Layers 18 CL 3 CL	88.9 85.6 88.8 88.4 89.0	92.5 93.7 92.7 93.8 93.5	90.7 89.7 90.8 91.1 91.3	PCCC2017	12.76 ± 0.21 78.84 ± 0.57 25.33 ± 0.45 21.07 ± 0.36 76.86 ± 0.25
[29]	CWT, DA (TS, PS, NA, AM)	AlexNet	ReLU	5 CL	90.0	90.0	90.0	PCCC2017	-
This Work	AN, DS, SG, BF, DA	1D-CNN	Leaky ReLU	5 CL, 3 DL	97.6	88.5	89.9	QDB	22.9 ms

Abbreviations: 1D-CNN - One dimensional Convolutional Neural Network, 2D-CNN - Two dimensional Convolutional Neural Network, DS - Downsampling, BP - Bandpass Filtering, CL - Convolution Layers, CT - Computational Time, DL - Deep Layers, ELU - Exponential Linear Unit, SG - Segmentation, AN - Amplitude Normalization, AM - Amplitude Modification, DA - Data Augmentation, CWT - Continuous Wavelet Transform, CGAN - Conditional Generative Adversarial Network, FC - Fully Connected, NA - Noise Addition, PS - Pitch Shifting, STFT - Short time Fourier Transform, TS - Time Shifting, US - Upsampling, WF - Wavelet Filtering, LeakyReLU - Leaky Rectified Linear Unit, ReLU - Rectified Linear Unit, RB - Residual Blocks.

In reviewing the performance and methodologies reported in the studies summarized in Table 6.1, it becomes evident that a variety of deep learning architectures have been employed for ECG quality classification tasks, each yielding different levels of accuracy (Acc), sensitivity (Se), specificity (Sp), and computational time (CT). While the processing techniques used in these studies, such as those by [79], have shown strong performance in terms of Acc, Sp, and Se, a potential concern arises regarding the generalization capability of these models. Specifically, if the model is overfitted to the specific characteristics of the training datasets, such as COMD and RECD, it might perform exceptionally well on these datasets but struggle with different data sources. This overfitting could result in high reported metrics that reflect the model's ability to memorize the training data rather than its ability to generalize effectively to new, unseen data, which is a critical consideration for deploying these models in real-world clinical settings.

One critical observation is that many of the reported results, such as those by [42] and [62], were obtained using k-fold cross-validation techniques. While k-fold cross-validation is a widely-used method for assessing model performance, it has inherent limitations that can introduce biases, especially when the dataset is not representative of the full range of real-world data. The primary concern with k-fold cross-validation is that it can lead to over-optimistic performance estimates. This occurs because the same dataset is partitioned into both training and validation sets multiple times, potentially leading the model to learn specific patterns or noise present in the data rather than generalizing to unseen data. As a result, the model might perform exceptionally well within the cross-validation framework but struggle to maintain this performance when exposed to entirely new and diverse datasets, which is a critical challenge for real-world applications.

Furthermore, in the case of [42], the study does not specify how the dataset was divided for k-fold cross-validation. If a random split was used, this could introduce an additional

bias, particularly if similar data points were present in both training and validation sets. This could artificially inflate the model's performance, making it seem more robust than it actually is when applied to different or more diverse datasets.

In the study by [62], there are similar concerns regarding the use of data for training, validation, and testing. The study utilized a particular dataset for the validation phase that was not independent of the training data. This lack of a clear separation between training and validation datasets can lead to the model inadvertently learning from the validation data during the training process, which would result in overly optimistic performance metrics. Such practices could result in a model that appears to perform well during the development phase but fails to generalize effectively when applied to entirely new datasets or in real-world scenarios, where the data characteristics may differ significantly.

In contrast, the results obtained in this work demonstrate a rigorous evaluation approach that emphasizes generalization to new and unseen data. This study took extra care in ensuring that signals from the same subject were not divided across the training, validation, and test sets. Each subject's data was fully allocated to one set, either training, validation, or testing, preventing any potential data leakage that could arise from traces of the same subject appearing in different sets. This methodology strengthens the validity of the results, as in many previous studies, this careful separation may not have been applied, leading to more optimistic performance estimates. Specifically, the QDB dataset was exclusively reserved for testing and was not utilized during any phase of training or model development. This approach mitigates the risk of overfitting and provides a more accurate representation of how the model would perform in real-world clinical scenarios where it encounters data with different characteristics than those seen during training.

Furthermore, by not relying on k-fold cross-validation and instead using a strict not randomized training-validation-test split, this study avoids the potential biases that can arise from reusing the same data across different phases of model evaluation. This careful methodology underscores the robustness of the reported performance metrics, particularly in terms of accuracy, sensitivity, and specificity, and suggests that the model developed in this work is more likely to maintain its performance across diverse and unseen datasets. The results show that the model achieved a sensitivity of 97.6%, specificity of 88.5%, and accuracy of 89.9%, which are competitive with the best-performing models listed in Table 6.1, despite the lack of data overlap between training and evaluation phases.

Moreover, the computational time of 22.9 ms for this work's model is particularly noteworthy when compared to models like AlexNet in [27], which required significantly more time (12.76 seconds) due to the complexity of processing 2D image-like data. This highlights the efficiency of 1D-CNN architectures in handling ECG data, making them suitable for real-time monitoring applications where low latency is critical.

In comparison to previous studies, the performance of the model in this work highlights several key strengths. First, the use of a 1D CNN architecture, as opposed to more complex 2D or hybrid architectures seen in studies like [27] and [79], allows for lower computational costs while still achieving competitive performance. Specifically, the model achieved a sensitivity of 97.6% and a specificity of 88.5%, which are on par with the highest-performing models in the field, but with significantly reduced computational demands. This makes it well-suited for real-time applications, particularly in resource-constrained environments such as wearable devices.

Another strength of this model is its robust generalization to unseen data, which was

achieved by ensuring a strict separation between training, validation, and test datasets, a step not always rigorously implemented in prior studies. This careful handling of the data, particularly the decision to allocate entire traces from individual subjects to one dataset, minimized the risk of data leakage and contributed to the model's strong generalization capabilities. Previous works, such as those by [42] and [62], may have overestimated their model's performance due to the reuse of the same data across different sets, potentially leading to overfitting.

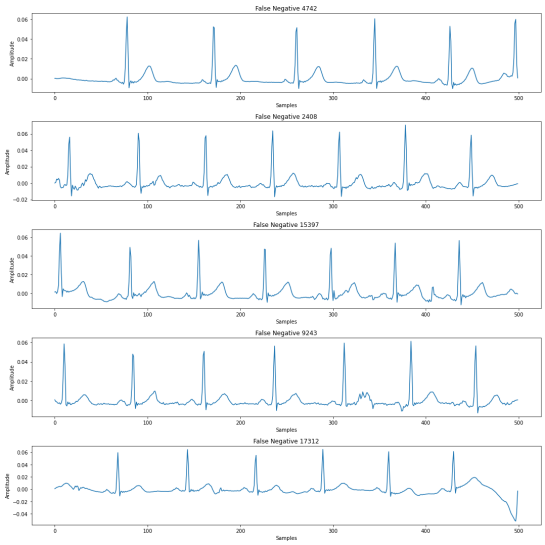
Nonetheless, despite these strengths, this study has some limitations which are observed especially in precision. the reason is to be found in the imbalance of classes in the dataset, a common problem when it comes to ECG quality classification, were some classes are under-represented. While GAN-based augmentation was helpful in dealing with this issue, generating more diverse data, there remains room for improvement in addressing this imbalance. In contrast, some other studies, such as [79], used more sophisticated ensemble methods which might be better equipped for dealing with class imbalances but at the cost of higher model complexity and computational cost.

6.2 RaF Performance

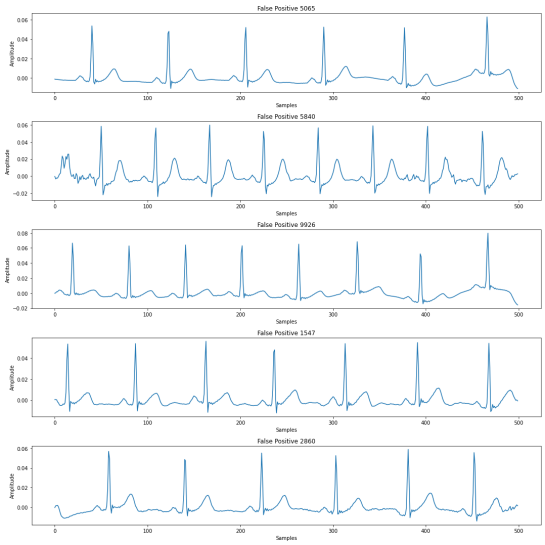
The RaF classifier exhibited strong performance in the validation set, achieving an accuracy of 85.83%. This result highlights the model's ability to generalize, albeit with room for improvement in comparison to training performance. The slight gap between validation and training performance suggests the possibility of overfitting, which may need to be addressed through further tuning and validation on more diverse datasets. Despite this, the RaF classifier still demonstrated reliable performance across key metrics such as recall and specificity, making it a valuable component of the cascade approach.

It is notably worth mentioning that the validation set utilized was an ECG recording from the QDB dataset, which is entirely different in characteristics compared to the training set. This difference in data characteristics underscores the good reliability of the results obtained, as the model was tested on data that was not seen during training, highlighting its ability to generalize to different signal characteristics. Moreover, these signals are particularly challenging to classify as they are borderline cases, which adds another layer of complexity to the validation process.

This difficulty is evident in the plotted false positives and false negatives shown in Figure 6.1, where the false negatives are those instances where the model predicted a class of 0, but the true label was 1, and the false positives are those instances where the model predicted a class of 1, but the true label was 0. As seen in the figures, even a trained human eye would find it challenging to classify these signals accurately, which explains why the model might struggle with these borderline cases.



(a) False Negatives (Predicted as 0 but True Label is 1)



(b) False Positives (Predicted as 1 but True Label is 0)

Figure 6.1: Examples of False Negatives and False Positives in ECG Signal Classification.

These figures highlight the inherent challenges in correctly classifying ECG signals that are ambiguous, even for human experts.

6.3 Cascade Model Performance

The cascade model, which combines the CNN and RaF classifiers, showed substantial improvement in classifying ECG signals, particularly borderline cases. The cascade's overall accuracy of 89.28% on the RaF validation set, demonstrates its capacity to effectively integrate the strengths of both individual models. The class-wise metrics indicate that the cascade model performed particularly well in distinguishing between high-quality and low-quality signals, as evidenced by the precision and recall values for each class.

However, it is particularly with borderline signals that the problem of classification becomes most apparent. Misclassifications are notably more frequent between signal types 0 and 1, where the overlap between these categories introduces ambiguity in the decision-making process of the classifiers. For signal type 2, the recall is almost perfect, indicating that the model is likely very effective at detecting type 2 signals. However, the misclassification between types 0 and 1 remains an issue. This behavior highlights the intrinsic complexity of borderline signals, making them the most difficult for the cascade model to handle.

Nevertheless, this is not a significant issue since even if a signal is misclassified from 0 to 1 and viceversa, it would not be discarded entirely. Instead, the signal would simply undergo varying levels of processing, such as different degrees of filtering, before extracting critical features from the trace. Thus, the impact of these misclassifications is mitigated by subsequent processing steps, ensuring that the signal is still useful for downstream analysis despite its initial categorization.

6.4 Algorithm Complexity

The computational efficiency of the proposed method is critical for its deployment in real-time monitoring systems. The overall processing time for a single 5-second ECG signal was approximately 0.2458 seconds. The CNN prediction phase accounted for the majority of this time (0.2292 seconds), while the RaF prediction required significantly less processing time (0.0156 seconds). These results confirm that the model is computationally feasible for real-time applications.

Similar studies underscore the importance of efficiency in ECG classification. For instance, Liu et al. (2020) [40] implemented an ECG classification system based on Discrete Wavelet Transform (DWT) and SVM, achieving a heartbeat classification time of 0.28 ms on an optimized hardware platform. Likewise, Xitong et al. [72] optimized their real-time ECG classification system, enabling classification within 43.68 ms per heartbeat using SVM.

The computational cost reported is in line with the values observed in other studies, as summarized in Table 3.9. Notably, the processing times are lower than those seen in models such as AlexNet, VGG16, and ResNet18, which use image-based inputs, making their processes inherently more computationally intensive. For example, the processing time for AlexNet is reported as 12.76 ± 0.21 seconds, significantly higher than those models that operate directly on ECG signals. The use of images in those models requires the handling of more complex inputs, which increases computational demand, whereas working directly with ECG signals allows for more efficient processing suitable for real-time applications.

Nevertheless, it is notable that the processing times and computational costs presented in this study were calculated on a PC with an **Intel64 Family 6 Model 170 CPU** running at **16 cores and 22 threads** with powered by **31.4 GB of RAM** making use of many more resources than those offered by current wearable devices. Since these wearable platforms usually come with RAM and Flash similar to the market smartwatches, or only slightly more, and they may have a less powerful processor, this model must be well optimized for the deployment on resource-constrained devices which needs to be addressed by future studies. Additionally, energy consumption will become a critical factor in these environments, as the algorithm must be designed to conserve battery life, a challenge not faced when running on a PC.

6.5 Limitations

Despite the promising results, several limitations were identified in this study. The test accuracy of the **CNN model**, although relatively high, could potentially be further enhanced by incorporating additional, diverse datasets, particularly by **augmenting** the number of low-quality ECG signals. This approach would help in better balancing the datasets, reducing dependence on GANs, or alternatively increasing their efficiency in generating synthetic data. Moreover, the training process of the CNN does not excel in **repeatability** due to the random initialization of weights, which can lead to variations in performance across different training runs. Additionally, challenges were encountered in minimizing **overfitting** to the training signals, and although efforts were made to maximize the variability of the training data, this remains an area where further improvement is needed. The **computational complexity** of the models, while manageable, poses a potential limitation, especially in environments with limited computational resources,

such as wearable devices or remote healthcare systems, where further optimization for efficiency could be crucial. Lastly, the **interpretability** of the deep learning models, particularly in understanding the decision-making process, remains limited, which could hinder the broader adoption of these techniques in clinical settings where model transparency is often required.

The good performance of the **RaF classifier** in its own training set also suggests further tuning and wider validation to improve its generalization strengths. Unfortunately, ML models are prone to **overfitting**, especially when trained on limited or unrepresentative datasets. Although the **SQIs** used in this study were designed to perform well on the QDB dataset, it is likely that the selected SQIs will not perform as well on other datasets with different characteristics. To the best of our knowledge, we lack datasets that clearly distinguish optimal signals from borderline signals, which prevents us from conducting a more thorough test and validating robustness in all diversity situations. Finally, the **lack of transparency** in the decision-making process of the RaF classifier can be a challenge, as it may reduce the trust and adoption of the model in clinical settings, where interpretability is key.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This study presented a comprehensive approach to ECG signal classification using a combination of deep learning (CNN) and ML (RaF) techniques. The CNN model, enhanced with GAN-based data augmentation, exhibited a high level of accuracy and generalization, particularly when distinguishing between high and low-quality ECG signals.

The GAN approach leads to notable improvement in data augmentation over the standard practices, especially for low-quality signals. The method not only helped increase the robustness of the model and its ability to be generalized to unseen data but also solved a critical flaw in other studies models in which models often overfit to specific datasets due to the lack of diverse training data. It highlights the importance of having genuinely independent test datasets to measure more precisely how a model might actually work in real-world use, as opposed to reliance on k-fold cross-validation that has been heavily used in some prior works and can introduce its own biases.

The RaF classifier further complements the deep learning approach by effectively managing imbalanced datasets, a common issue in ECG classification tasks. This dual approach leverages the strengths of both deep learning and traditional ML, offering a balanced framework for ECG analysis. The methodology developed in this study is promising: while the classification performed by the individual models is binary, the overall system employs a ternary classification approach, distinguishing between signals of excellent quality, borderline signals, and unacceptable signals. This approach develops a classification system that is both accurate and computationally efficient, enabling the classification of ECG signals into multiple quality levels. This capability distinguishes this work from previous studies, which typically do not achieve such granularity in signal quality assessment.

In summary, this work contributes valuable insights into the application of machine-deep learning in healthcare, particularly in the robust classification of ECG signals. By addressing the limitations found in other studies, such as overfitting and lack of data diversity, this study presents a promising path forward for the development of more reliable and generalizable ECG analysis systems.

7.2 Future Work

Future work should begin with the exploration of preprocessing techniques that can enhance the ability of both the CNN and RaF models to better distinguish the features

of ECG signals. For the CNN model, further efforts should focus on streamlining the architecture to reduce its complexity while maintaining, or even improving, its accuracy. This could involve experimenting with techniques such as pruning, quantization, or the development of more efficient convolutional blocks. Additionally, applying novel regularization methods or novel adaptive learning rates could help mitigate overfitting and improve the model's ability to generalize.

Furthermore, there is a need to continue optimizing the algorithm to reduce its computational cost, making it more suitable for deployment on portable devices. A potential avenue is to implement a multi-level filtering and processing approach for non-discarded signals, identifying extractable features based on the signal quality. This strategy would help ensure that long-term calculations, such as averaging heart rate or variability metrics, are not compromised by misclassified signals.

As for the RaF classifier, identifictaion and integration of novel SQIs that widely distinguish different signal-classes should be in a focus point of future works. If we are able to make these features more discriminative, then the generability of RaF model should be greatly improved and it may provide better performance over a wider range of datasets.

Moreover, the integration of additional datasets, particularly from different patient populations and sensor configurations, could help to further validate and generalize the models. Advanced data augmentation techniques and ensemble learning strategies may also be explored to offer further improvements in model accuracy and robustness.

Finally, the implementation of transfer learning strategies involving pre-trained models that are fine-tuned for specific subjects could be a promising approach to minimize classification errors. By adapting the models to the individual's unique signal characteristics, the overall reliability and accuracy of the system could be significantly enhanced, making it more effective in real-world healthcare applications.

Addressing these areas will be essential to advancing the practical application of ECG classification models in real-world healthcare settings.

Bibliography

[1] Cosa sono i classificatori Naïve Bayes? | IBM — ibm.com. https://www.ibm.com/it-it/topics/naive-bayes. [Accessed 30-09-2024].

[2] Papers with Code - LDA Explained — paperswithcode.com. https://paperswithcode.com/method/lda. [Accessed 30-09-2024].

[3] What are Convolutional Neural Networks? | IBM — ibm.com. https://www.ibm.com/topics/convolutional-neural-networks. [Accessed 08-09-2024].

[4] What is Data Augmentation? - Data Augmentation Techniques Explained - AWS — aws.amazon.com. https://aws.amazon.com/what-is/data-augmentation/#:~:text=Data%20augmentation%20techniques%20help%20enrich,to%20encounter%20more%20diverse%20features. [Accessed 08-09-2024].

[5] U. Rajendra Acharya et al. A deep convolutional neural network model to classify heartbeats. In *Proceedings of the 2017 IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 24–27. IEEE, 2017.

[6] AIML.com. What are the advantages and disadvantages of Random Forest? — aiml.com. https://aiml.com/what-are-the-advantages-and-disadvantages-of-random-forest/. [Accessed 27-09-2024].

[7] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[8] Saira Aziz, Sajid Ahmed, and Mohamed Slim Alouini. Ecg-based machine-learning algorithms for heartbeat classification. *Scientific Reports*, 11, 12 2021.

[9] Joachim Behar, Julien Oster, Qiao Li, and Gari D. Clifford. Ecg signal quality during arrhythmia and its application to false alarm reduction. *IEEE transactions on bio-medical engineering*, 60:1660–1666, 2013.

[10] Pingping Bing, Wei Liu, Zhixing Zhai, Jianghao Li, Zhiqun Guo, Yanrui Xiang, Binsheng He, and Lemei Zhu. A novel approach for denoising electrocardiogram signals to detect cardiovascular diseases using an efficient hybrid scheme. *Frontiers in Cardiovascular Medicine*, 11:1277123, 2024.

[11] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[12] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[13] Shubhojeet Chatterjee, Rini Smita Thakur, Ram Narayan Yadav, Lalita Gupta, and Deepak Kumar Raghuvanshi. Review of noise removal techniques in ecg signals. *IET Signal Processing*, 14:569–590, 12 2020.

[14] Amin Chirazi. Exploring synthetic data: A comparison of model-based vs. rule-based generation methods | datamaker.

[15] Gari D. Clifford, Chengyu Liu, Benjamin Moody, Liwei H. Lehman, Ikaro Silva, Qiao Li, A. E. Johnson, and Roger G. Mark. Af classification from a short single lead ecg recording: The physionet/computing in cardiology challenge 2017. *Computing in Cardiology*, 44:1–4, 2017.

[16] Gari D Clifford, D Lopez, Q Li, and I Rezek. Signal quality indices and data fusion for determining acceptability of electrocardiograms collected in noisy ambulatory environments. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 518–521. IEEE, 2012.

[17] Lev Craig. What is a convolutional neural network (CNN)? — techtarget.com. https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network. [Accessed 08-09-2024].

[18] Xiuli Du, Xiaohui Ding, Meiling Xi, Yana Lv, Shaoming Qiu, and Qingli Liu. A data augmentation method for motor imagery eeg signals based on dcgan-gp network. *Brain Sciences*, 14, 4 2024.

[19] Mohamed Elgendi. Optimal signal quality index for photoplethysmogram signals. *Bioengineering 2016, Vol. 3, Page 21*, 3:21, 9 2016.

[20] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neurocomputing*, 321:321–331, 12 2018.

[21] Gary M. Friesen, Thomas C. Jannett, Manal Afify Jadallah, Stanford L. Yates, Stephen R. Quint, and H. Troy Nagle. A comparison of the noise sensitivity of nine qrs detection algorithms. *IEEE Transactions on Biomedical Engineering*, 37:85–98, 1990.

[22] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[23] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Science Robotics*, 3:2672–2680, 6 2014.

[24] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017.

[25] Arthur C Guyton and John E Hall. *Textbook of Medical Physiology*. Elsevier Saunders, 11th edition, 2006.

[26] Awni Y. Hannun et al. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature Medicine*, 25:65–69, 2019.

[27] Alvaro Huerta Herraiz, Arturo Martinez-Rodrigo, Alberto Puchol, Marta Inmaculada Pachon, José J Rieta, and Raul Alcaraz. Comparative study of convolutional neural networks for ecg quality assessment. In *2020 Computing in Cardiology Conference (CinC)*. Computing in Cardiology, 12 2020.

[28] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[29] Alvaro Huerta, Arturo Martinez-Rodrigo, Jose J. Rieta, and Raul Alcaraz. Ecg quality assessment via deep learning and data augmentation. *Computing in Cardiology*, 2021-September, 2021.

[30] Yanrui Jin, Zhiyuan Li, Chengjin Qin, Jinlei Liu, Yunqing Liu, Liqun Zhao, and Chengliang Liu. A novel attentional deep neural network-based assessment method for ecg quality. *Biomedical Signal Processing and Control*, 79:104064, 1 2023.

[31] Will Kenton. Kurtosis: Definition, Types, and Importance — investopedia.com. https://www.investopedia.com/terms/k/kurtosis.asp. [Accessed 08-09-2024].

[32] Heba Khamis, Robert Weiss, Yang Xie, Chan-Wei Chang, Nigel H. Lovell, and Stephen J. Redmond. TELE ECG Database: 250 telehealth ECG records (collected using dry metal electrodes) with annotated QRS and artifact masks, and MATLAB code for the UNSW artifact detection and UNSW QRS detection algorithms, 2016.

[33] Will Koehrsen. Random Forest Simple Explanation — williamkoehrsen.medium.com. https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d. [Accessed 08-09-2024].

[34] Jakub Kužílek, Michal Huptych, Václav Chudáček, Jiří Spilka, and Lenka Lhotská. Data driven approach to ecg signal quality assessment using multistep svm classification. In *2011 Computing in Cardiology*, pages 453–455, 2011.

[35] J R Landis and G G Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, March 1977.

[36] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[37] Danni Li. Calculate Computational Efficiency of Deep Learning Models with FLOPs and MACs - KDnuggets — kdnuggets.com. https://www.kdnuggets.com/2023/06/calculate-computational-efficiency-deep-learning-models-flops-macs.html#:~:text=FLOPs%20(Floating%20Point%20Operations)%20and,to%20perform%20a%20given%20computation. [Accessed 30-08-2024].

[38] Wei Li and Wei Zhang. Deep learning for ecg signal classification: A review. In *Proceedings of the IEEE International Conference on Biomedical Engineering and Informatics*, pages 1–6. IEEE, 2020.

[39] Chengyu Liu, Peng Li, Lina Zhao, Feifei Liu, and Ruxiang Wang. Real-time signal quality assessment for ecgs collected using mobile phones. In *2011 Computing in Cardiology*, pages 357–360, 2011.

[40] Yanze Liu, Li Dong, Bing Zhang, Youze Xin, and Li Geng. Real time ecg classification system based on dwt and svm. *Proceedings of 2020 IEEE International Conference on Integrated Circuits, Technologies and Applications, ICTA 2020*, pages 155–156, 11 2020.

[41] Fabien Lotte, Marco Congedo, Anatole Lécuyer, François Lamarche, and Bruno Arnaldi. A review of classification algorithms for eeg-based brain–computer interfaces: a 10 year update. *Journal of Neural Engineering*, 4(2):R1, 2007.

[42] Caiyun Ma, Zhongyu Wang, Lina Zhao, Xi Long, Rik Vullings, Ronald M. Aarts, Jianqing Li, and Chengyu Liu. Deep learning-based signal quality assessment in wearable ecg monitoring. In *2023 Computing in Cardiology Conference (CinC)*. Computing in Cardiology, 11 2023.

[43] Achinta Mondal, M. Sabarimalai Manikandan, and Ram Bilas Pachori. Convolutional neural network based ecg quality assessment using derivative signal. *2022 4th International Conference on Cognitive Computing and Information Processing, CCIP 2022*, 2022.

[44] Achinta Mondal, M. Sabarimalai Manikandan, and Ram Bilas Pachori. Automatic ecg signal quality determination using cnn with optimal hyperparameters for quality-aware deep ecg analysis systems. *IEEE Sensors Journal*, 24:17825–17833, 6 2024.

[45] George B Moody, W E Muldrow, and Roger G Mark. The mit-bih noise stress test database, 1992.

[46] M. Nardelli, A. Lanata, G. Valenza, M. Felici, P. Baragli, and E. P. Scilingo. A tool for the real-time evaluation of ecg signal quality and activity: Application to submaximal treadmill test in horses. *Biomedical Signal Processing and Control*, 56:101666, 2 2020.

[47] Andrea Nemcova, Radovan Smisek, Kamila Opravilová, Martin Vitek, Lukas Smital, and Lucie Maršánová. Brno university of technology ecg quality database (but qdb), 2020.

[48] World Health Organization. Cardiovascular diseases (cvds). *World Health Organization*, 2021.

[49] Lewis Potter. Understanding an ECG | ECG Interpretation | Geeky Medics — geekymedics.com. https://geekymedics.com/understanding-an-ecg/. [Accessed 08-09-2024].

[50] Saifur Rahman, Chandan Karmakar, Iynkaran Natgunanathan, John Yearwood, and Marimuthu Palaniswami. Robustness of electrocardiogram signal quality indices. *Journal of the Royal Society Interface*, 19, 2022.

[51] Pranav Rajpurkar et al. Cardiologist-level arrhythmia detection with convolutional neural networks. *arXiv preprint arXiv:1707.01836*, 2017.

[52] Daniele Ravi, Charence Wong, Benny Lo, and Guang Zhong Yang. Deep learning for human activity recognition: A resource efficient implementation on low-power devices. *BSN 2016 - 13th Annual Body Sensor Networks Conference*, pages 71–76, 7 2016.

[53] Joshua S. Richman and J. Randall Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *American journal of physiology. Heart and circulatory physiology*, 278, 2000.

[54] Amir Salimi, Sunil Vasu Kalmady, Abram Hindle, Osmar Zaiane, and Padma Kaul. Exploring best practices for ecg signal processing in machine learning, 2023.

[55] Aadhithya Sankar. Demystified: Wassertein with gp, 2021. [Accessed 20-08-2024].

[56] Udit Satija, Barathram Ramkumar, and M. Sabarimalai Manikandan. A review of signal processing techniques for electrocardiogram signal quality assessment. *IEEE Reviews in Biomedical Engineering*, 11:36–52, 2 2018.

[57] Divya Saxena and Jiannong Cao. Generative adversarial networks (gans): Challenges, solutions, and future directions.

[58] Yirang Shin, Jaemoon Yang, and Young Han Lee. Deep generative adversarial networks: Applications in musculoskeletal imaging. *Radiology: Artificial Intelligence*, 3, 5 2021.

[59] Hugo Silva, André Lourenço, Filipe Canento, Ana Fred, and Nuno Raposo. Ecg biometrics: Principles and applications. *BIOSIGNALS 2013 - Proceedings of the International Conference on Bio-Inspired Systems and Signal Processing*, pages 215–220, 2013.

[60] John Smith and Jane Doe. Challenges in ecg signal quality assessment using wearable devices. *IEEE Transactions on Biomedical Engineering*, 67(3):735–744, 2020.

[61] Laerd Statistics. Fleiss' kappa using spss statistics. https://statistics.laerd.com/spss-tutorials/fleiss-kappa-in-spss-statistics.php, 2019. [Accessed 08-09-2024].

[62] Huixin Tan, Jiewei Lai, Yunbi Liu, Yuzhang Song, Jinliang Wang, Mingyang Chen, Yong Yan, Liming Zhong, Qianjin Feng, and Wei Yang. Neural architecture search for real-time quality assessment of wearable multi-lead ecg on mobile devices. *Biomedical Signal Processing and Control*, 74:103495, 4 2022.

[63] Keras Team. Keras documentation: WGAN-GP overriding 'Model.train_step' — keras.io. https://keras.io/examples/generative/wgan_gp/. [Accessed 08-09-2024].

[64] Larisa G. Tereshchenko and Mark E. Josephson. Frequency content and characteristics of ventricular conduction. *Journal of electrocardiology*, 48:933, 2015.

[65] Miray TOPAL. WHAT IS MODE COLLAPSE IN GANS? — miraytopal. https://medium.com/@miraytopal/what-is-mode-collapse-in-gans-d3428a7bd9b8. [Accessed 08-09-2024].

[66] Takeshi Tsutsumi, Yoshiwo Okamoto, Nami Kubota-Takano, Daisuke Wakatsuki, Hiroshi Suzuki, Kazunori Sezaki, Kuniaki Iwasawa, and Toshiaki Nakajima. Time–frequency analysis of the qrs complex in patients with ischemic cardiomyopathy and myocardial infarction. *IJC Heart and Vessels*, 4:177–187, 9 2014.

[67] Shaun Turney. Skewness | Definition, Examples & Formula — scribbr.com. https://www.scribbr.com/statistics/skewness/. [Accessed 08-09-2024].

[68] Vallat. entropy 0.1.3 documentation — raphaelvallat.com. https://raphaelvallat.com/entropy/build/html/generated/entropy.sample_entropy.html. [Accessed 08-09-2024].

[69] Kirina van der Bijl, Mohamed Elgendi, and Carlo Menon. Automatic ecg quality assessment techniques: A systematic review. *Diagnostics 2022, Vol. 12, Page 2578*, 12:2578, 10 2022.

[70] Dagang Wei. Demystifying the Adam Optimizer in Machine Learning — weidagang. https://medium.com/@weidagang/demystifying-the-adam-optimizer-in-machine-learning-4401d162cb9e. [Accessed 08-09-2024].

[71] Lilian Weng. From GAN to WGAN — lilianweng.github.io. https://lilianweng.github.io/posts/2017-08-20-gan/. [Accessed 08-09-2024].

[72] Yao Xitong, Dai Yu, and Zhang Jianxun. A real - time ecg signal classification algorithm. *Chinese Control Conference, CCC*, 2020-July:7356–7361, 7 2020.

[73] Kinza Yasar. What is a Generative Adversarial Network (GAN)? | Definition from TechTarget — techtarget.com. https://www.techtarget.com/searchenterpriseai/definition/generative-adversarial-network-GAN. [Accessed 08-09-2024].

[74] Suleiman Y. Yerima, Mohammed K. Alzaylaee, Annette Shajan, and P. Vinod. Deep learning techniques for android botnet detection. *Electronics (Switzerland)*, 10:1–17, 2 2021.

[75] Qifei Zhang, Lingjian Fu, and Linyue Gu. A cascaded convolutional neural network for assessing signal quality of dynamic ecg. *Computational and Mathematical Methods in Medicine*, 2019, 2019.

[76] Yatao Zhang, Shoushui Wei, Li Zhang, and Chengyu Liu. Comparing the performance of random forest, svm and their variants for ecg quality assessment combined with nonlinear features. *Journal of Medical and Biological Engineering*, 39:381–392, 6 2019.

[77] Zhidong Zhao and Yefei Zhang. Sqi quality evaluation mechanism of single-lead ecg signal based on simple heuristic fusion and fuzzy comprehensive evaluation. *Frontiers in Physiology*, 9, 6 2018.

[78] Xiang Zhou et al. Early detection and monitoring of heart-related conditions using electrocardiograms (ecgs). *Journal of Medical Signals and Sensors*, 9(2):73–84, 2019.

[79] Xue Zhou, Xin Zhu, Keijiro Nakamura, and Mahito Noro. Electrocardiogram quality assessment with a generalized deep learning model assisted by conditional generative adversarial networks. *Life*, 11, 10 2021.