NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF PHYSICS

MÁRCIA INÊS ALMEIDA MONTEIRO

BSc in Biomedical Engineering

ASSESSING ELECTROCARDIOGRAM QUALITY: A DEEP LEARNING FRAMEWORK FOR NOISE DETECTION AND CLASSIFICATION

MASTER IN BIOMEDICAL ENGINEERING

NOVA University Lisbon
September, 2024

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
PHYSICS

ASSESSING ELECTROCARDIOGRAM QUALITY: A DEEP LEARNING FRAMEWORK FOR NOISE DETECTION AND CLASSIFICATION

MÁRCIA INÊS ALMEIDA MONTEIRO

BSc in Biomedical Engineering

Adviser: Hugo Filipe Silveira Gamboa
Full Professor, NOVA University Lisbon

MASTER IN BIOMEDICAL ENGINEERING

NOVA University Lisbon
September, 2024

Assessing Electrocardiogram Quality: A Deep Learning Framework for Noise Detection and Classification

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.1.18) [1].

Acknowledgements

Primeiramente, gostaria de agradecer ao Professor Hugo Gamboa por esta oportunidade e por despertar em mim o interesse pela área de deep learning. À Mariana Dias, estou eternamente agradecida, pois, sem o teu suporte incondicional, não teria conseguido fazer isto sozinha. A tua experiência e conhecimento foram cruciais. Obrigada por planeares comigo estas semanas e por sempre acreditares que era possível. "Nós" fizemos e entregamos esta tese juntas. Agradeço também a todos os membros do LIBPhys pelo ambiente colaborativo e acolhedor desde o primeiro dia. Muito obrigada ao Luís, Philip, Dânia, Inês e Rodrigo.

Muito obrigada às minhas colegas que partilharam esta caminhada comigo, nomeadamente à Leonor, pela imensa companhia e por todas as conversas motivacionais. Ao meu braço direito e amante, a minha Sara, por toda a ajuda, conversas e suporte. Sem ti, nunca teria acabado esta jornada. Desde mudanças de casa a problemas pessoais e académicos, estiveste sempre ao meu lado, e sei que continuarás a estar. Por isso, um grande obrigada do fundo do meu coração. Serás sempre uma segunda casa para mim. Quem diria que iria de Invicta à capital cobiçar a mais linda de Pombal.

À minha amiga Jéssica, sempre tão carinhosa e que me viu crescer ao longo destes cinco anos, não podia pedir alguém melhor do que tu. Obrigada por todas as cadeiras partilhadas, refeições e risos. À minha Alexandra, que sempre caía mas se levantava, agradeço por seres a pessoa que mais me fez rir incondicionalmente. Obrigada por estares ao meu lado durante todo este tempo e por me permitires acompanhar-te ao longo da tua caminhada.

À minha família, que este feito não seria possível sem eles, agradeço profundamente. Confiaram e "bancaram" a minha vinda para Lisboa, e a eles dedico esta tese. Pai, obrigada por, ao longo destes anos, me teres aturado com todos os comboios que perdi por distração, bilhetes mal tirados e viagens sem bilhete. Obrigada por me deixares ir contigo na cabine do maquinista e por todos os momentos cómicos e bifanas de graça. Mãe, obrigada por todas as refeições providenciadas e os "chocolatinhos da motivação". Sem essas pequenas coisas, não teria tido a energia para continuar. À minha irmã "stinky", obrigada por me ouvires sempre e por acreditares em mim. Todas as nossas conversas e a tua vontade

de estar envolvida na minha vida significam o mundo para mim. Estou eternamente orgulhosa de ti Rute, por partilharmos esta jornada académica desde o infantário e, agora, tenho um imenso orgulho em ver-te a entregar a tua tese ao meu lado.

À minha Lilas, obrigada também pelas horas ao telefone, que fortaleceram o nosso laço, que tão facilmente poderia ter-se perdido. Serei eternamente grata e mal posso esperar pela tua vez de acabares a faculdade. Tens o meu apoio incondicional. Muito obrigada à minha Braz, que sempre continuou a ser uma figura importante na minha vida e uma grande amiga, apesar da distância de Lisboa.

Por fim, ao Pedro, meu companheiro de vida e fonte constante de força. Obrigada por me apoiares incondicionalmente, por me ouvires em cada momento de frustração e por seres o meu equilíbrio em todas as situações. A tua presença nesta jornada foi essencial, e sou imensamente grata por ter-te ao meu lado.

Abstract

The electrocardiogram (ECG) is essential for diagnosing cardiovascular conditions, yet ECG signals are highly prone to noise, which reduces their reliability, especially in wearable devices and long-term monitoring applications. Traditional noise detection methods achieve limited accuracy, and deep learning (DL) has emerged as a promising solution. Current DL approaches are focused on binary classification (noisy versus clean) and lack more detailed quality information. This paper explores these challenges by developing a DL model capable of performing a comprehensive signal quality assessment (SQA) through the detection of noisy segments and the classification of specific noise types in the ECG data. To accomplish this, a labeled dataset is generated by injecting controlled noise into clean signals. A model based on gated recurrent units (GRUs) is developed to handle the time-series nature of the data, identifying and classifying common noise types. The model achieved a high accuracy of 92.86 % for electrode motion (EM) noise followed by an accuracy of 92.05 % and 81.55 % for baseline wander (BW) and muscle artifacts (MA), respectively. Despite challenges with overlapping noise, the model consistently maintains high recall and precision rates, proving effective in identifying noise when present. Compared to state-of-the-art DL approaches, the proposed model significantly outperforms previous methods, achieving 99.72 % accuracy when conducting a binary classification of the signals.

This thesis offers a significant advancement in ECG signal processing by providing a more detailed understanding of noise beyond binary classification. Future improvements comprise adding the quantification of noise to predict the level of contamination. The work has the potential to impact clinical practice, long-term monitoring, and self-monitoring technologies, benefiting medical professionals, researchers, and wearable device users alike.

Keywords: Electrocardiogram, Signal Quality Assessment, Deep Learning, Detection, Classification, Gate Recurrent Units, Baseline Wander, Muscle Activation, Electrode Motion, Wearables

Resumo

O eletrocardiograma (ECG) é uma ferramenta diagnóstica crucial para doenças cardiovasculares. No entanto, os sinais de ECG são vulneráveis a ruído, que comprometem a sua qualidade e utilidade, especialmente em monitorização a longo prazo e dispositivos Vestíveis. Os métodos tradicionais de deteção de ruído apresentam extatidão limitada, e o deep learning (DL) surge como uma solução promissora. As abordagens atuais de DL concentram-se na classificação binária (ruído versus sinal limpo), sem fornecer informações detalhadas sobre a qualidade dos sinais. Esta tese enfrenta esse desafio desenvolvendo um modelo de DL que realiza uma avaliação abrangente da qualidade do sinal (SQA) ao detetar segmentos ruidosos e classificar tipos específicos de ruído nos dados de ECG.

Foi criado um conjunto de dados anotado ao injetar ruído controlado em sinais limpos. Para lidar com a natureza temporal dos dados, foi desenvolvido um modelo baseado em gated recurrent units (GRUs), capaz de identificar e classificar os tipos de ruído mais comuns. O modelo alcançou uma extatidão de 92.86 % na deteção de movimento de eletrodos, 92.05 % para ruído de deriva de linha de base (BW) (MA) e 81.55 % para artefactos musculares (MA). Embora a sobreposição de ruídos apresente desafios, o modelo mantém altas taxas de recall e precisão, demonstrando eficácia na deteção de ruído quando presente. Comparado com as abordagens mais recentes, o modelo proposto supera métodos anteriores, atingindo 99.72 % de extatidão na classificação binária dos sinais.

Este trabalho representa um avanço significativo no processamento de sinais de ECG, oferecendo uma compreensão mais detalhada do ruído para além da classificação binária. Melhorias futuras incluem a quantificação do ruído para prever o nível de contaminação. O trabalho tem potencial para impactar a prática clínica, a monitorização a longo prazo e a automonitorização, beneficiando profissionais de saúde, investigadores e utilizadores de dispositivos vestíveis.

Palavras-chave: Eletrocardiograma, Avaliação da Qualidade do Sinal, Deep Learning, Deteção, Classificação, Gate Recurrent Units, Deriva de Linha de Base, Ativação Muscular, Movimento do Elétrodo, Dispositivos Vestíveis

Contents

List of Figures

List of Tables

List of Abbreviations

Adam	Adaptive Moment Estimation *(p. 23)*
AN	Artificial Neuron *(p. 6)*
ANN	Artificial Neural Network *(p. 6)*
bas	Baseline Relative Power *(pp. 27, 32, 48)*
BCEWithLogitsLoss	Binary Cross-Entropy with Logits Loss *(p. 22)*
BW	Baseline Wander *(pp. 4, 5, 14, 17, 19, 20, 22, 23, 25, 26, 29–31, 35, 36, 39)*
CD	Conduction Disturbance *(p. 17)*
CGAN	Conditional Generative Adversarial Network *(pp. 13, 15)*
CNN	Convolutional Neural Network *(pp. 12, 13, 15)*
DL	Deep Learning *(pp. 2, 6, 7, 12–14, 37–40)*
ECG	Electrocardiogram *(pp. 1, 2, 4, 5, 10–21, 28, 37–40)*
EM	Electrode Movement *(pp. 4, 5, 11, 14, 17, 20, 22, 23, 25, 26, 29–31, 35–37, 39)*
FC	Fully Connected *(pp. 6, 12, 15, 22, 28)*
FN	False Negatives *(pp. 25, 26, 35, 36)*
FNN	Feedforward Neural Network *(pp. 6–8)*
FP	False Positives *(pp. 24–26, 35, 36, 39)*
GPU	Graphics Processing Unit *(p. 16)*
GRU	Gated Recurrent Units *(pp. 8, 9, 21, 22, 24, 28, 37, 39)*
HYP	Hypertrophy *(p. 17)*
ICU	Intensive Care Unit *(p. 1)*

ID Identification *(p. 19)*

kurt Kurtosis *(pp. 27, 32, 48)*

LMS Least Mean Squares *(p. 11)*
LSTM Long Short-Term Memory Neural Network *(pp. 8, 9, 13, 15)*

MA Muscle Activation *(pp. 4, 5, 14, 17, 20, 22, 23, 25, 26, 29–31, 35–37, 39)*
MI Myocardial Infraction *(p. 17)*
ML Machine Learning *(p. 5)*

NORM Normal *(p. 17)*

PCA Principal Component Analysis *(p. 11)*
PLI Powerline Interference *(pp. 4, 11)*
psd Power Spectral Density *(pp. 27, 32, 48)*

RNN Recurrent Neural Networks *(pp. 7, 8, 13)*
ROC Receiver Operating Characteristic *(p. 24)*
RRM Residual Recurrent Module *(pp. 13, 15)*

skew Skewness *(pp. 27, 32, 48)*
SNR Signal-to-Noise Ratio *(pp. 27, 32, 37, 39, 48)*
SQA Signal Quality Assessment *(pp. 2, 3, 5, 11, 12, 14, 15, 37, 39)*
SQI Signal Quality Indicator *(pp. 10, 11, 26–28, 48)*
STTC ST-T Changes *(p. 17)*

TN True Negatives *(pp. 25, 26)*
TP True Positives *(pp. 24–26)*

WFDB Waveform Database Software *(p. 16)*

1 Introduction

Cardiovascular diseases stand as the predominant cause of mortality worldwide, claiming approximately 17.9 million lives annually [2]. This alarming statistic highlights the importance of effective diagnostic methods. In this context, Electrocardiogram (ECG) is a crucial tool, providing a real-time record of the heart's electrical activity. However, despite its widespread use in clinical and personal health monitoring the ECG is highly susceptible to interference and artifacts, usually denominated as noise, which can severely degrade the quality of the recordings, sometimes rendering them unusable [3]. These challenges persist even in controlled clinical settings, such as 12-lead resting or stress tests, where noise can still occur, sometimes requiring the exam to be repeated.

In the context of long-term monitoring, the adversities posed by noise become increasingly evident. Arrhythmias, characterized by brief and irregular episodes of heart activity, need extended monitoring over several days to achieve accurate diagnoses [4]. This type of monitoring can be performed in Intensive Care Units (ICUs), as well as with wearble devices such as the Holter [5]. Holter devices enable long-term monitoring of patients at home, users are often advised to avoid strenuous activities or showering, as the device is not designed to withstand water exposure or physical stress from exercise. Such conditions may interfere with its functionality, potentially leading to inaccuracies in monitoring and gaps in the detection of transient arrhythmias.

Wearables, such as ECG patches [6], are also widely used in a sports settings, where real-time monitoring of physiological parameters is essential for tracking performance and avoiding increased risk for CVDs [7]. In addition, wearable devices can be employed to monitor workers [8] in order to adjust the work schedules, settings, and tasks with the aim to promote occupational health. Self-monitoring [9] plays an increasingly bigger role in today's society for personal health management. However, these contexts, much like day-to-day activities, introduce significant levels of noise into the recorded data, which can complicate the accuracy of the collections. This flexibility comes with challenges, as it often leads to higher noise levels in the recorded ECG data. An additional application of these devices in a research context is their usage as a source of data for datasets.

Consequently, there is an urgent need for more sophisticated noise identification systems to enhance signal quality assessment (Signal Quality Assessment (SQA)). Traditional methods represent an earlier attempt at performing SQA, but their reliance on global thresholds limits their effectiveness [10, 11]. While rule-based approaches account for some variations, they still depend heavily on fixed threshold values, reducing their applicability. In contrast, deep learning (Deep Learning (DL)) methodologies offer significant improvements, as they generalize better by learning important features associated with ECG signals and distinguishing noisy signals with high accuracy. However, a key limitation of these approaches is their focus on binary classification of signals as either clean or noisy [12], which may not be ideal when only segments of the signal are affected or when tailoring denoising methods based on the type of noise present would be more effective.

The growing need for noise-robust ECG monitoring systems presents as the central motivation for this thesis. This project introduces a DL classifier and detector capable of identifying noisy segments in ECG recordings and categorizing the type of noise present. By addressing the need for more detailed noise analysis, this approach could open new possibilities for self and long-term ECG monitoring. In this context, both medical professionals and individuals engaged in self-monitoring could potentially benefit from the integration of alert systems, enabling timely corrective actions when necessary. The classifier also plays a crucial role in the broader context of ECG signal processing as it generates a detailed evaluation of noisy intervals, making it a valuable resource for both researchers and users. This comprehensive classification guides the selection of appropriate denoising methods, whether employing straightforward filters or more sophisticated techniques, and also facilitates the creation of large noise-labeled datasets.

In summary, this thesis aims to provide a solution that expands on the traditional binary classification of clean versus noisy ECG signals, offering a more detailed understanding of the noise present in the data. The work presented has the potential to benefit a diverse range of users, including clinical staff, researchers, individuals self-monitoring their health, and those interested in signal processing, paving the way for new opportunities in both clinical applications and future research.

1.1 Objectives

The primary objective of this thesis is to develop a robust method for the quality assessment of ECG signals by classifying noise. To achieve this, the main objective can be divided into four specific sub-goals:

1. Develop and evaluate a deep learning model capable of distinguishing between different noise types and identifying the intervals of corruption.

2. Evaluate the performance of the developed model in a binary classification context (noisy or clean)

3. Compare with state of the art methods

By accomplishing these sub-objectives, the thesis aims to significantly advance SQA methodologies and be applicable in ambulatory and clinical settings.

This thesis is structured to first define key concepts that are frequently referenced throughout the document. It then provides a review of the state of the art, assessing current methods in the field. Following this, the methodology used in the research is presented, leading into the results and a discussion of the findings, before concluding with final remarks.

The research work described in this dissertation was carried out in accordance with the norms established in the ethics code of Universidade Nova de Lisboa. The work described and the material presented in this dissertation, with the exceptions clearly indicated, constitute original work carried out by the author.

2

Theoretical Concepts

The chapter will comprise of essential theoretical concepts used thorough the thesis.

2.1 The Electrocardiogram

The ECG is a diagnostic tool that measures the electrical activity of the heart. It provides a visual representation of the heart's function, offering crucial insights into rhythm, conduction, and potential irregularities [13]. The test involves placing electrodes on the chest to detect changes in electrical potentials, which are then recorded and displayed as waveforms. These changes correspond to the depolarization and repolarization processes in the heart muscle. Key components of the ECG include the P wave, indicating atrial depolarization; the QRS complex, representing ventricular depolarization; and the T wave, which reflects ventricular repolarization, marking the heart's recovery phase before the next beat.

A lead in ECG terminology refers to a specific view of the heart's electrical activity [13]. A 12-lead ECG provides 12 different views, which allow for a three-dimensional understanding of the heart's electrical activity, similar to looking at an object from different angles revealing its full shape and structure. Wearable devices such as smartwatches, patches [6] and Holter [5] monitors provide simpler methods for heart monitoring by using a reduced lead system compared to the traditional 12-lead ECG. These devices may use a single-lead ECG resembling Lead I, a two-lead configuration similar to Leads I and II or I and III, or modified chest leads that approximate the precordial leads (V1-V6) in a simplified form.

2.1.1 Types of Noise Found in ECGs

Noise in the context of ECG signals refers to any unwanted electrical artifacts that distorts the true representation of the heart's electrical activity by deforming the typical features of the waveform. In ECG signals, various types of noise, such as Powerline Interference (PLI), Baseline Wander (BW), Electrode Movement (EM), and Muscle Activation (MA), are present. This project will concentrate specifically on baseline wander,

electrode movement, and muscle activation. Powerline interference [14], characterized by a sinusoidal waveform of 50 or 60 Hz from the alternating current power supply, can be easily filtered due to its isolated frequency.

BW is a low-frequency noise, typically around 0.5 to 1 Hz. It is caused by factors such as respiration or body movements [15]. The noise causes the ECG baseline to drift up and down, rather than remaining steady [14]. The magnitude of the drift can exceed the amplitude of the QRS affecting the ECG analysis.

MA noise originates from electrical activity of the surrounding skeletal muscles and can be triggered by sudden movements. MA [15] noise usually overlaps significantly with the frequency components of the QRS (above 100 Hz), ranging from 0.01 to 100 Hz [14, 16] this makes it difficult to distinguish between muscle activity and heart activity.

EM occurs as a result of poor electrode contact with the skin, leading to changes in the impedance of the interface. These artifacts usually have a frequency range of 1 to 10 Hz and can be caused by patient movement and incorrect electrode placement [14]. EM noise can often manifest as large-amplitude waveforms in the ECG signal, which can be mistaken for QRS complexes, thus distorting the accurate interpretation of heart electrical activity.

2.2 Signal Quality Assessment

SQA in the context of ECGs is the process of evaluating the accuracy and reliability of electrocardiogram signals. It involves detecting and quantifying noise, artifacts, and other distortions that can affect the clarity of the heart's electrical activity [17]. Effective SQA enables verification of clean ECG, enhancing the quality and usefulness of the signal for various applications.

2.3 Machine Learning

Machine Learning (ML) is enables computers to learn from experience without being explicitly programmed for specific tasks. Instead of following predefined instructions, a machine learning model is provided with data and uses it to identify patterns and make predictions [18]. As the model processes more data, it improves its ability to understand and make accurate predictions, effectively learning how to perform tasks independently by analyzing examples. Supervised learning is a specific approach of ML in which a model learns from labeled data [18]. The model is provided with input-output pairs, learning to map inputs to the correct outputs, labels by analyzing examples. Over time, as it is exposed to more labeled data, the model improves its ability to make accurate predictions on new, unseen data, effectively learning how to perform the task by being guided with the correct answers during training.

2.4 Deep Learning

DL, a subfield of machine learning that uses multi-layered neural networks to model complex data patterns [19]. Initial layers extract basic features from data, while deeper layers build more complex representations, reflecting higher-level cognitive processes [20].

An Artificial Neuron (AN) can be seen as analogous to a human neuron, serving as the fundamental unit of an Artificial Neural Network (ANN), just as neurons are the core components of human nervous system. Artificial neurons can be thought of as decision-making units, capable of communicating with each other in a manner similar to biological neurons.

Although the functioning of the human brain is much more complex and far from being fully understood, the following simplified parallel can be drawn. In both artificial and biological systems, neurons receive signals or data as inputs, each with an assigned weight that reflects its importance [21], this parallelism is illustrated in figure 2.1.In artificial neurons, weights are numerical values that adjust inputs, determining each input's contribution to the neuron's output by increasing or decreasing its relevance [21, 22]

In ANs, such as Feedforward Neural Networks (FNNs)—the most fundamental type of neural networks—the neuron sums the inputs after adjusting them based on their weights. The summed inputs are then passed through an activation function (Fig. 2.1) [21], which determines if the signal should be transmitted to other neurons based on whether it exceeds a certain threshold. This process is crucial for the network's decision-making. The activation function introduces non-linearity into the model, allowing the FNN to learn from data to handle complex tasks. After the activation function processes the input, the final output can either serve as an input for subsequent neurons or act as the model's final prediction.

In a Fully Connected (FC) layer (also known as a linear layer), all neurons from the preceding layer are connected to every neuron in the current layer [23], allowing for comprehensive interactions between neurons across adjacent layers. This structure is fundamental in many neural networks and helps to integrate information across different layers effectively.

The equation for the output of a neuron in a neural network can be represented as shown in Equation 2.1.

$$y_{w_i,b} = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{2.1}$$

Where $y_{w_i,b}$ represents the output, w_i are the weights, x_i are the inputs, b is the bias, and f is the activation function. The bias [24] functions as a baseline adjustment, shifting the entire output uniformly up or down, regardless of the inputs. Essentially, it acts as a constant offset added to the weighted sum of the input features.

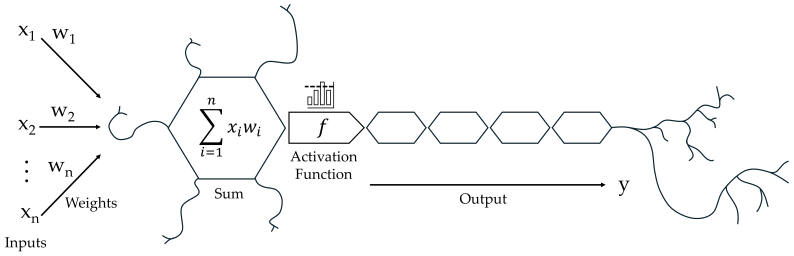


Figure 2.1: Artificial neuron overview

2.4.1 The Training Process

In a similar way that humans learn through study and practice, the training process in DL involves the model learning from data to perform a specific task. During supervised learning, this means using a dataset with known outputs to teach the model. The model's performance is evaluated by comparing its predictions to these known results, with the goal of minimizing training error and ensuring the model generalizes well to new, unseen examples. Just as humans refine their understanding through repeated learning and feedback, the FNN improves its accuracy through iterative training and adjustments [25].

Backpropagation [25] is used to propagate errors backwards through the network, adjusting the parameters at each layer to reduce the error. These adjustments are guided by the gradients of the loss function, which indicate the necessary changes. Optimization [25] plays a central role in this process, involving the minimization or of the loss function (also referred to as the cost function, objective function, or error function).

Optimizers are crucial for implementing the adjustments based on the gradients. They determine how the parameters are updated [23, 25], employing various strategies to influence the speed and effectiveness of the learning process. The choice of optimizer can significantly impact how quickly and effectively the model converges and its final performance.

Training DL models presents several challenges. Overfitting happens [25] when the model learns the training data too well, including specific details that do not generalize to new, unseen data. This can be mitigated through regularization techniques such as dropout and cross-validation. Additionally, imbalanced datasets, where certain classes or features are underrepresented, can lead to biased models. Addressing this requires strategies such as adding penalties to the parameters of the loss function [25].

2.4.2 Architectures of Neural Networks

In neural network design, different architectures are suited to different tasks, especially as complexity increases. While FNNs are effective for straightforward input-to-output mappings, they are limited in handling sequences or dynamic data.

Recurrent Neural Networkss (RNNs) are motivated by the simplicity of FNNs but are designed to process sequential data by allowing information to persist across time steps.

Unlike FNNs, which only consider the current input, RNNs have loops that enable the network to retain information from previous inputs, making them suitable for tasks where context or memory is essential.

2.4.2.1 Long Short Term Memory Networks

This workflow in RNNs [23] is particularly useful in tasks like language modeling or time series prediction, where the current output depends not just on the current input but also on previous inputs. Long Short-Term Memory Neural Networks (LSTMs) build upon this idea by introducing mechanisms to better manage the flow of information across longer sequences, addressing issues like vanishing gradients that standard RNNs may face.

LSTMs [26, 27] are an extension of RNNs with the addition of input, output, and forget gates. These gates control the flow of information through the cell. The forget gate decides what information from the previous cell state should be discarded. The input gate determines what new information should be stored, using the current input and the previous hidden state to decide which values to update. The output gate controls what information should be passed to the next hidden state, based on the current input and the previous hidden state [23]. These gates allow LSTMs to control and manipulate information over long sequences effectively.

2.4.2.2 Gated Recurrent Networks

Although LSTMs are powerful, they are also complex due to their multiple gates. To simplify this architecture, Gated Recurrent Unitss (GRUs) [27] were introduced, they merge the cell state and hidden state. They are also a variant of RNNs that combine the

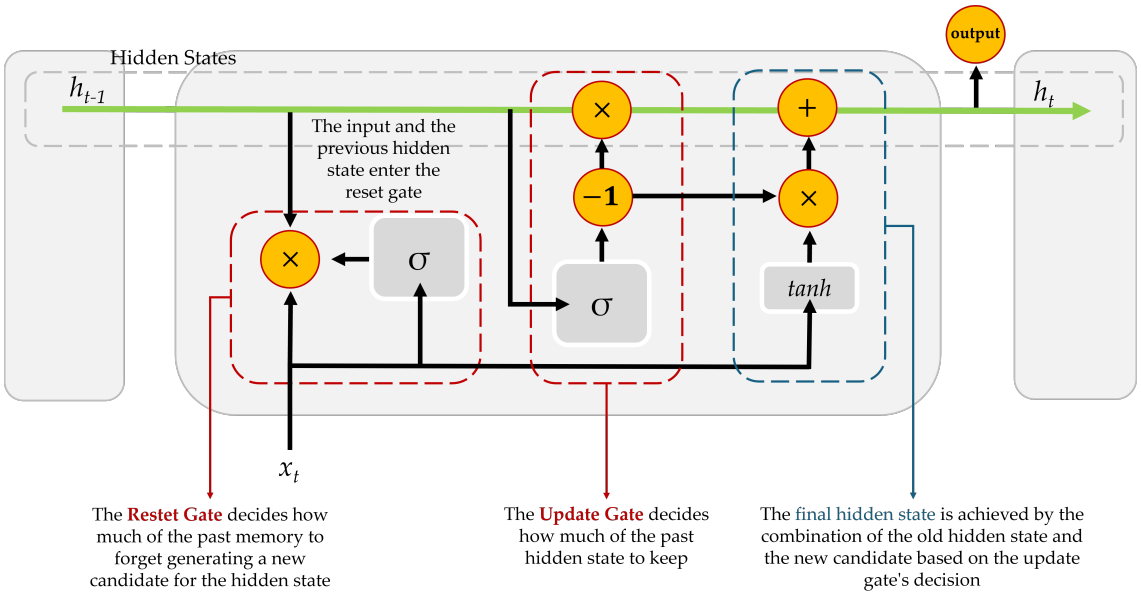


Figure 2.2: GRU overview

input and forget gates into a single update gate, which decides what information to keep from the previous time step and what new information to add, as illustraded in Figure 2.2. GRUs also include a reset gate, which determines how much of the past information to forget [23]. GRUs do not have an output gate, which simplifies their architecture while still maintaining the ability to manage long-term dependencies effectively. This makes GRUs computationally more efficient than LSTMs [28, 29], while often achieving comparable performance.

3

State Of The Art

The primary goal of ECG quality assessment is to identify noise that could distort critical features of the signal. By detecting and flagging these compromised signals, the process ensures that only clean and reliable data are used for further analysis. This literature review will first cover traditional methods, which typically depend on manually crafted features and fixed thresholds, and then shift focus to deep learning approaches that provide more flexible and scalable alternatives.

3.1 Traditional Metrics

Traditional methods for assessing the quality of ECG signals have evolved over time, employing a variety of approaches to evaluate the noise levels present in ECG recordings. Each of the following methods has its own set of techniques and focus on different features of the ECG signals. These can be divided in fundamental categories: statistics-, feature-, frequency-, morphology-based.

Statistical approaches focus on computing statistical metrics to check if the signal's distribution and shape falls under acceptable ranges that characterize noise-free signals. Sungho Oh's approach [30] incorporates various statistical metrics: (1) signal variance, where higher values suggest the presence of noise, (2) zero-crossing rate, which measures how frequently the signal transitions from positive to negative, (3) turn counts, which track the number of local minima that exceed 0.1 mV. Kurtosis [11, 31] and skewness [10, 11] have also been proposed as good indicators: these check for any anomalies in the signal's distribution, namely identifying abnormal sharpness and asymmetry, respectively. A significant challenge with these metrics is the need to establish appropriate thresholds and ranges that may not always be applicable in different contexts.

In 2018, Zhao [11] introduced a rule-based classification method that uses several metrics to assess ECG signal quality. This approach integrates traditional feature extraction with fuzzy logic to evaluate signals. The process is divided into two main steps. First, key Signal Quality Indicators (SQIs) are derived from the signal, including R peak detection, power spectral distribution, R-R interval variability, kurtosis, skewness, and

baseline power. These features are then combined through heuristic fusion, a rule-based approach that selects the optimal combination of SQIs based on accuracy. In the second step, a comprehensive fuzzy evaluation is applied. Fuzzy logic is useful when signal quality is not easily classified as simply "good" or "bad" assigning degrees of membership to the categories "excellent", "acceptable", or "poor". This method's dependence on heuristics and subjective parameter tuning can reduce its effectiveness, increasing the risk of misclassification in more complex or ambiguous cases.

Feature-based approaches assess signal quality by analyzing specific characteristics of the ECG. An example is the adaptive threshold QRS detection method proposed by Chiarugi et Al. [32]. In their approach, they utilize ECG features to calculate a noise index based on both the baseline level of the ECG and the variability of the QRS complex. This approach can have some limitations such as the correct estimation of the baseline level of the signal and the variability of the QRS complex. Sungho Oh [30] adopts a feature extraction and reduction approach for ECG SQA. His methodology employs Principal Component Analysis (PCA) to reduce the dimensionality of ECG data, distinguishing between significant features like heartbeats and noise. PCA helps in isolating noise by identifying components with lower eigenvalues, though its success depends on precise component selection.

Frequency-based SQA involves analyzing the frequency bands of the recorded signals, as the typical frequency range of an ECG is highly distinctive. In Liping Li's paper [33], these characteristic frequency bands are utilized. This method focuses on analyzing the power spectrum of the signal within the 0.05 to 30 Hz range, where most ECG features are concentrated, and compares it to the 30 to 60 Hz range, which is primarily associated with noise. By calculating this ratio, a quantitative measure of the noise level is obtained. That said, it is limited to addressing specific noise types, such as PLI EM noise, and also relies on static thresholding.

One foundational type of ECG SQA are the morphology-based methods, with Wang's work [34] being a prominent example. This approach evaluates signal quality by analyzing the differences in the area between successive QRS complexes and accumulating mismatches in a histogram. Although effective in highlighting discrepancies in the morphology of the ECG, this method is highly dependent on accurate QRS detection, which can be problematic if the noise present significantly distorts the signal. Additionally, it assumes a normal QRS complex morphology, potentially overlooking variations caused by pathological conditions. Morphological approaches rely on standard "clean" ECG as templates to compare real signals with and look for the presence of noisy components. Iravanian's [35] approach involves averaging the ECG signal and subtracting this average from the original signal to isolate the residual components. This residual can then be analyzed to estimate the noise present. However, this method relies on the assumption that the average ECG signal is representative of a clean signal, which may not always be the case. Rio et. Al [31] creates a template using Least Mean Squares (LMS) adaptive filtering to isolate the ECG component from the noise. However, this method's effectiveness is

heavily dependent on the quality and representativeness of the clean template.

In conclusion, while traditional methods provide valuable frameworks for ECG SQA, they each face inherent limitations. These methods often rely on predefined threshold values and assumptions about signal characteristics, which may not be universally applicable across different contexts or patient populations. Furthermore, the effectiveness of these metrics depends on the accuracy of signal features such as QRS detection or baseline estimation, which can be compromised in noisy environments. Therefore, while traditionaL SQA is important, it requires careful implementation and may benefit from adaptive approaches to account for the variability in real-world ECG recordings.

3.2 Deep Learning

DL methods for ECG SQA have evolved significantly over the past decade, offering increasingly sophisticated techniques for evaluating signal quality. Unlike traditional methods that depend on handcrafted features and expert-defined rules, DL models learn high-level features directly from ECG signals, allowing for more adaptive and scalable approaches. These advancements are particularly noticeable when comparing various architectures, datasets, and performance metrics, as summarized in Table 3.1, which provides a structured comparison of the models discussed.

Zhou et al. present an early example of a 1D Convolutional Neural Network (CNN) model [36] trained on the PhysioNet/CinC 2011 and 2017 datasets [37–39], achieving 94.30 % accuracy by classifying single-lead ECG signals as either acceptable or unacceptable. As shown in Table 3.1, the architecture, consisting of two convolutional layers followed by a FC layer, demonstrates that even simple CNN architectures can outperform traditional methods.

While Zhou's early CNN model excelled in handling single-lead signals, Huerta et al. [40, 41] expanded this approach by using scalograms and transfer learning with advanced image classification models to better manage noisy signals. These models include AlexNet, VGG16, and GoogLeNet. As highlighted in Table 3.1, AlexNet achieved the highest accuracy at 91.20 %, followed by VGG16 at 89.65 % and GoogLeNet at 90.75 %, underscoring the effectiveness of CNNs in frequency-domain noise detection.

Mondal's work [42] continued the exploration of CNN-based models, focusing on binary classification of ECG quality. In this approach, signal segments were corrupted with synthetically added noise, similar to the method used in this thesis. The architecture, consisting of three 1D CNN layers, pooling layers, and a dense layer. The method used the first-order derivative of the ECG to enhance high-frequency components typically associated with noise. This CNN-based model achieved a 91.77 % accuracy on the PhysioNet Challenge 2017 [38, 39] dataset.

Liu et al. [43] introduced a dual-input method, where the primary input was a scalogram, and the second input comprised handcrafted statistical features such as baseline drift and R-peak count. The CNN architecture, consists of three convolutional layers,

fused CNN-extracted features with handcrafted ones, effectively distinguishing between acceptable and unacceptable signals. However, the reliance on scalograms limited the model's applicability to raw ECG signals, where time-series architectures, such as LSTM models, may be more suitable.

J. Zhang et al. [44] tackled the temporal dependencies of ECG signals by utilizing LSTM structures. They developed a large private dataset with 179,130 records—one of the largest used in this field. Their LSTM-ECG model achieved 93.50 % accuracy by merging the features learned by LSTM layers with domain-specific features, including spectral distribution and waveform variation. Without these domain-specific features, the precision fell to 91.10 %, highlighting the efficacy of the dual input method.

DL approaches are highly data-dependent, and the limited size of public ECG databases can lead to misleading performance outcomes. This data limitation drives the need for data augmentation. Zhou et al. [45] addressed this by introducing a Conditional Generative Adversarial Network (CGAN), performing both data augmentation and quality assessment. The CGAN's generator, consisting of two LSTM layers, and the discriminator, composed of two CNN layers, generated artificial ECG segments to balance the dataset and improve training efficiency. The CGAN-based system achieved accuracies of 97.10 % and 96.40 % on two datasets, highlighting the role of data augmentation in improving model performance.

A more recent innovation is the use of attention mechanisms. Jin et al. [46] introduced the DAC-LSTM model, which combined CNNs and bidirectional LSTMs with attention mechanisms to enhance feature selection from 12-lead ECGs. CNNs and LSTMs extracted features, followed by a time-based attention mechanism to select the most important segments for classification, ending with a softmax classifier. The model achieved 94. 00 % accuracy, which makes it highly applicable in real-world clinical settings, such as triage. Similarly, Zhong et al. [47] incorporated attention mechanisms through Squeeze-and-Excitation modules within a DenseNet architecture, achieving a high accuracy of 96.02 %. However, while these approaches enhance feature selection and classification performance, they still fall short in characterizing noise in detail, particularly in identifying noise sources.

Chen et al. [48] introduced SwinDAE, a model combining a denoising autoencoder with a 1D Swin Transformer, designed to efficiently handle long ECG recordings while reducing computational complexity. The encoder used the Swin Transformer to break the ECG into patches and extract key features like the P wave, QRS complex, and T wave, filtering out noise. The model, trained with three loss functions, achieved an overall F1 score of 83.58 %, with precision at 97.62 % and sensitivity at 95.38 %, demonstrating its effectiveness in distinguishing signal quality levels.

X. Zhang et al. [49] developed a model for wearable ECGs using Residual Recurrent Modules (RRMs), combining CNNs and RNNs with residual connections. This architecture, tested on ECG data from 20 cardiovascular patients and the China Physiological Signal Challenge 2020 dataset [50], achieved 98.72 % accuracy for two-category classification and 92.31 % for three-category classification ("good", "medium", "poor"). However,

its reduced sensitivity to electrode motion artifacts remained a challenge.

Traditional ECG quality assessment methods rely on rigid empirical thresholds or statistical calculations, making them difficult to adapt to new datasets. DL offers flexibility and improved accuracy by learning directly from data, but these models typically require large labeled datasets, which are often hard to obtain.

Despite recent advances, current DL models, while effective in classification, fail to identify the specific types of noise present.

This thesis addresses these gaps by proposing a model that not only detects and classifies noise but also identifies the specific noise source, percentages, and temporal intervals of contamination within the ECG signal. Focuses on three types of noise, BW, MA, and EM, to provide a comprehensive quality assessment based on the overall presence of noise over time. By employing a unique vectorized representation, where each position corresponds to a specific noise type, this model provides a more robust solution compared to existing methods. Its capability to simultaneously detect and classify multiple noise types addresses critical gaps in current ECG SQA research.

Table 3.1: Overview of main attributes and results of state of the art studies

Ref.	Model	Architecture	Datasets	Classes	Acc %	Recall %	Spe %	Pre %	F1 %
[36]	1D CNN	2 CNN → 2 FC	PCCC11, PCC17	Clean/ Noisy	94.30	91.30	95.50	—	—
[40]	AlexNet + Scalogram	5 CNN/5 MaxPool → 3 FC	PCCC17	Clean/ Noisy	91.20	100.00	90.30	—	—
[41]	VGG16 + Scalogram	13 CNN/5 MaxPool → 3 FC → 1 Softmax	PCCC17	Clean/ Noisy	89.65	85.60	93.70	—	—
	GoggLeNet + Scalogram	2 CNN → 3 MaxPool → 9 Inception Modules → 1 AvgPool → 1 FC → 1 Softmax + 2 AuxClassifiers			90.75	88.80	92.70	—	—
[42]	1D CNN-dECG	3 CNN → 3 MaxPool → Flatten → 1 FC	PTB, MITBIHA, PCCC17	Clean/ Noisy	91.77	—	—	—	—
[43]	HC-Stats + Spectrogram	2 Inputs (Spec + Stats): 3 CNN/3 MaxPool → Concatenate → 1 FC → 1 Softmax	PCCC11	Clean/ Noisy	93.09	97.67	77.33	93.67	84.72
[44]	LSTM-ECG	3 LSTM → Merge (LSTM + Domain) → 1 FC → 1 Softmax	PCCC11, Private	Clean/ Noisy	93.50	81.20	97.20	—	—
	LSTM NF-ECG	3 LSTM → 1 FC → 1 Softmax			91.10	79.20	94.60	—	—
[45]	CGAN -SQA	CGAN: 2 LSTM (G) → 2 CNN (D) → 1 FC → 1 Sigmoid SQA: 2 CNN → 2 LSTM → 1 FC	PCCC17, TELE ECG, MITBIHA, MIT-BIHNSR	Clean/ Noisy	97.10 — 96.40	98.60 — 99.10	96.40 — 95.00	—	—
[46]	DAC-LSTM	Channel Attention → 5 CNN → 1 BiLSTM → Time Attention → Softmax	PCCC11	Clean/ Noisy	94.00	97.59	76.47	—	—
[47]	DenseNet-SE	DenseNet (CNN) → SE Attention	PCCC11	Clean/ Noisy	96.02	99.19	86.16	—	—
[48]	SwinDAE	1D Swin Encoder → 1D CNN Decoder → 1 FC	PTB-XL, BUT QDB	High/ Medium /Low Quality	—	95.38	—	97.62	83.58
[49]	RRM	3 RRM: (Recurrent + Residual → 1D CNN) → Softmax	CPSC2020	Clean/ Noisy + High/ Medium/ Low Quality	97.72 2 labels) 91.74 (3 labels)	—	—	—	—

The mentioned datasets are available at: PhysioNet [39] Challenge 2011 (PCCC11) [37]; PhysioNet [39] Challenge 2017 (PCCC17) [38]; TELE ECG dataset (TELE ECG) [51]; MIT-BIH Arrhythmia Database (MITBIHA) [39, 52]; MIT-BIH Normal Sinus Rhythm Database (MIT-BIHNSR [39, 53]; Physikalisch-Technische Bundesanstalt Database (PTB) [39, 54]; PTB-XL Database (PTB-XL) [39, 55]; Brno University of Technology ECG Quality Database (BUT QDB) [56]; China Physiological Signal Challenge 2020 (CPSC2020) [50]. The abreviations used are: Max Pooling (MaxPool); Average Pooling (AvgPool); Auxiliary Classifier (AuxClassifier); Handcrafted (HC); Spectogram (Spec), Statistic (Stat); Non-Features (NF); Generator (G); Discriminator (D); Dual Attentional Convulutional (DAC); Squeeze-and-Excitation (SE); Swin Transform (Swin); Denoising Autoencoder (DAE).

4

Methods

This chapter covers the methods applied in the development of the model alongside with the computational resources, programming environment and datasets used.

4.1 Computational Resources and Programming Environment

This project was developed using Python [57] language. The model development was carried out using Pytorch [58], with the computational power provided by an NVIDIA RTX 6000 Ada Generation [59] Graphics Processing Units (GPUs) card. For data manipulation and visualization, the primary packages employed were pandas [60], numpy [61], Waveform Database Software (WFDB) [39, 62] and matplotlib [63]. The entire workflow was managed within Pycharm [64].

4.2 Public Datasets

As outlined in the Introduction, this thesis involves the generation of a custom dataset, of ECGs with controlled injections of typical ECG noise. To generate the custom dataset, two datasets were used, these are available in PhysioNet [39], a public repository of physiological. The datasets selected were the PTB-XL [55, 65] and the MIT-BIH [66].

4.2.1 PTB-XL PhysioNet Dataset

The PTB-XL ECG dataset [55, 65] is a large-scale publicly accessible collection of 21,837 12-lead clinical ECG recordings, each lasting 10 seconds, gathered from 18,885 patients. The dataset is stored in a 16-bit binary format with a resolution of 1 μV/LSB and provides recordings in two formats: the original high-resolution version with a 500 Hz sampling frequency and a down-sampled version at 100 Hz. PTB-XL includes metadata on signal quality, addressing issues like noise, baseline drifts, static noise, burst noise, and electrode problems.

PTB-XL is particularly notable for its diversity, encompassing a wide variety of ECGs, including Normal (NORM), Conduction Disturbance (CD), Hypertrophy (HYP), Myocardial Infraction (MI), and ST-T Changes (STTC) diagnostics. The distribution of these categories is shown in Table 4.1. The dataset consists of 56.36 % normal ECGs and 43.64 % of pathological ECGs. The annotations were performed by cardiologists, with peer review by a second cardiologist in some cases, ensuring high precision.

Table 4.1: Summary of ECG records by diagnostic category

Number of Records	Diagnostic	Description
9514	NORM	Normal ECG
5469	MI	Myocardial Infarction
5235	STTC	ST/T Changes
4898	CD	Conduction Disturbance
2649	HYP	Hypertrophy

Overall, PTB-XL provides a comprehensive resource for advancing research in automated ECG analysis, offering extensive annotations, high-quality signals, and detailed metadata.

4.2.2 MIT-BIH Noise Stress Test PhysioNet dataset

To create a noisy version of the ECGs, noise was overlaid on the clean ECG signals. The MIT-BIH Noise Stress Test Database was utilized for this purpose, specifically selecting records associated with EM, BW and MA [55, 66]. These types of noise were chosen due to their prevalence in ECG recordings and their significant impact on signal quality, allowing a controlled but realistic simulation of noisy ECGs.

The noise dataset used contains three half-hour noise recordings that are typically found in ambulatory ECG recordings. These recordings were obtained using physically active volunteers, with standard ECG recorders, leads, and electrodes that were placed to ensure that the recording predominantly captured noise rather than ECG signals.

The noise recordings were sampled at 360 Hz and consist of two channels. The three noise records include EM, BW and MA.

4.3 Generating Custom Dataset

This thesis requires the process of generating noisy controlled signals from clean ECGs, this process is illustrated in Figure 4.1.

4.3.1 PTB-XL and MIT-BIH Records Curation

The first step involves loading and preparing the PTB-XL ECG Dataset along with the MIT-BIH Noise Stress Test Database. The PTB-XL dataset is first downloaded from PhysioNet, and the associated metadata is processed to eliminate records with noise filtering out records with annotations indicating issues such as baseline drift, static noise,

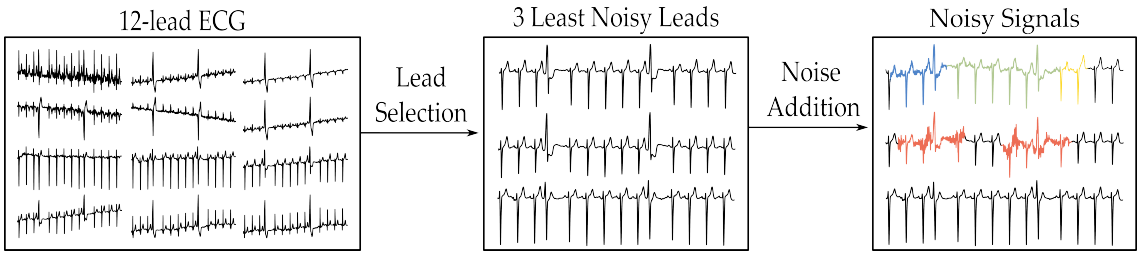


Figure 4.1: Generation of noisy signals (signal 10100 of the test set)

burst noise, or electrode problems, ensuring only high-quality ECG signals are retained. The prepared clean ECGs and noise records are stored locally in an accessible format, ready for subsequent processes.

4.3.2 Pre-processing

Prior to creating the noisy ECG records for the custom dataset, a few preprocessing steps were applied to ECG records from the database. This corresponds to the first step of the pipeline illustrated in Figure 4.1.

1. **Resampling:** Each record is resampled from its original frequency (500 Hz) to 360 Hz. This resampling aligns the ECG signals with the noise data of the MIT-BIH database.

2. **Min-max scaling:** The amplitude of the signal was normalized [67] between 0 and 1 this improves the convergence speed of the model, and prevents any single feature from dominating the learning process due to its scale.

3. **Filtering:** The records are filtered using a combination of a bandpass filter and moving average [68], effectively removing baseline drifts and smoothing the signal without compromising the integrity of the signal itself.

    a) The bandpass has a range of 1 to 45 Hz [68] isolating the relevant ECG frequencies.

    b) The moving average uses a sliding window of size 7 to compute the average of neighboring values in the ECG signal, smoothing out very rapid fluctuations and reducing high-frequency noise while preserving key features such as ECG peaks. Padding is applied by mirroring the first and last 3 samples to prevent edge artifacts and ensure smooth convolution at the boundaries.

4. **Lead Selection:** The filtered records undergo a lead selection phase where R-peaks and total peaks (including R-peaks and other peaks) are analyzed. Leads with fewer than 8 R-peaks are excluded to ensure that empty signals are discarded. Among the remaining leads, the three with the fewest number of total peaks are selected, as a higher number of non-fiducial peaks is indicative of higher noise. Therefore, the selected leads are considered less noisy and prioritized.

This pipeline (Figure 4.1) ensures that only the cleanest signals are used, retaining just three leads per record while maintaining a controlled environment for the noise injection.

4.3.3 Train, Test and Validation Split

The train-test-validation split of the ECGs is done according to the patient's Identification (ID) to ensure that the data from a patient does not appear in more than one subset. This patient-level splitting is fundamental to avoid any data leakage, which could lead to bias. The data set is then divided accordingly, 70 % of the patient IDs are allocated to the training set, 15 % to the validation set, and the remaining 15 % to the test set. This split by patients IDs results in different total numbers of ECG in the test and validation sets, as seen in Table 4.2. The same 70/15/15 split is applied to normalized noise records.

Table 4.2: Number of records in train, validation and testing sets

Set	Number of Records
Training	35157
Validation	7509
Testing	7592

4.3.4 Model's Inputs

This subsection aims to simulate real noisy ECG signals by injecting noise typically associated with ECG data into the clean signals processed in Section 4.3.2, this step is also depicted in Figure 4.1. This ensures that the data used as input to the model is diverse enough to emulate the signals encountered in real-world scenarios. The method applied involves adding noise to the clean signals based on the following principles:

1. **Number of Noise Intervals:** The number of noise intervals to be added is generated randomly, ranging from 0 to a specified maximum (set to 4 due to the duration of the signals being 10 seconds). The inclusion of 0 as a possible outcome ensures that some signals are kept clean, mimicking real-world variability

2. **Noise Types and Interval Selection:**

   a) BW noise has a minimum interval length of 5 seconds (1800 samples) to simulate prolonged disturbances, typical of baseline wander.

   b) Other noise types have a maximum interval length of 5.6 seconds (2000 samples), ensuring that the noise does not dominate the entire signal.

3. **Smoothing the Noise Transitions:** Smooth transitions at the beginning and end of the noise intervals are applied using a moving average filter to avoid abrupt changes that could create unrealistic noisy signals. This step simulates more natural occurring noise patterns.

4. **Scaling the Noise:** The noise is also scaled using a random factor between 0.2 and 1, ensuring variability.

5. **Saving the Noisy Signals and Noise Information:** Both the noisy signals and the detailed information about the noise intervals are saved. The noise information includes the starting and ending samples of the noise addition, along with the one-hot encoding of the noise types present per interval. This logging is essential for generating the true labels.

4.3.5 Model's Outputs

The model's output is a one-hot encoded vector with the same length as the input signal. Each position of the one-hot encoded array represents a type of noise, in this case MA, EM, BW. The structure of the output is shown in Figure 4.2. The generation of the true labels handles overlapping intervals, allowing multiple types of noise to be present simultaneously at any given time.

The one-hot encoded labels generated by this method provide a clear and structured representation of the noise present in each signal, which is crucial for training models to detect and classify different types of noise. The method's ability to handle overlapping noise intervals reflects real-world scenarios where multiple noise types may coexist.

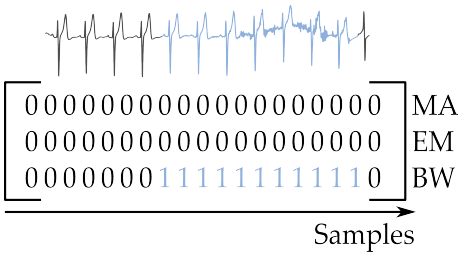


Figure 4.2: Example of the model's output (signal 265 of the test set)

4.4 Model Design

In this thesis, the main structure of the neural network is designed to detect and classify different types of noise in ECG signals. The model outputs a one-hot encoded vector, representing the presence of each type at each time step. This section details the model architecture, including its structure and components. It also covers the processes for training, validation, and testing.

4.4.1 Model Architecture

4.4.1.1 Input Sequence

The model accepts input sequences of the size ['batch_size', 'sequence_length', 'input_size']. Where the 'input_size' is 1, as each time step corresponds to a single value from the ECG. In this context the input sequence has the following shape: ['batch_size', 3600, 1]. The overall structure of the model is displayed in Figure 4.3

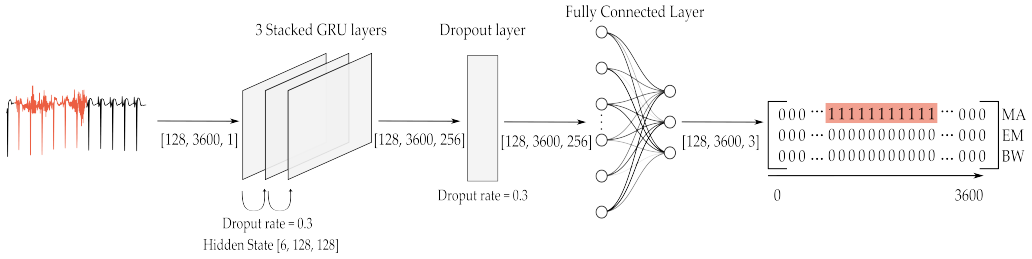


Figure 4.3: Final model overview and output interpretation: The input tensor has a shape of [128, 3600, 1], where 128 is the batch size, 3600 is the sequence length, and 1 represents the input size (number of features per time step). The hidden state in the GRU is [6, 128, 128], reflecting 3 stacked layers with bidirectional processing (3×2 directions), where the first 128 is the batch size and the second 128 is the hidden size (number of neurons). After processing through the GRU, the output has a shape of [128, 3600, 256] (due to being a bidirectional stack), which is passed a dropout layer and through a fully connected layer, reducing the dimensions to [128, 3600, 3] to classify each time step into one of 3 possible states. (Signal 150 of the test set)

4.4.1.2 Layers

1. **GRU Layers:** Stacked GRU layers process the input sequence and produce an output sequence of hidden states, each hidden state corresponding to a time step in the input sequence. The length of the output sequence remains the same as the input sequence, preserving the temporal structure of the data. These layers are configured with the following parameters:

   a) `hidden_size`: Defines the size of the hidden state vector at each time step. The hidden `'hidden_size'` corresponds to the number of neurons in the layer. This results in an output tensor of shape [`batch_size`, 3600, `'hidden_size'`]

   b) `bidirectional`: This parameter can be set to either to `'True'` - processing the input sequence in both directions, effectively doubling the hidden size for each time step, though this does not change the sequence length - or `'False'`- processing the input sequence in a single direction, maintaining the original hidden size. This approach may capture less contextual information compared to previous setting.

   c) `dropout`: Dropout defines the rate at which a random subset of hidden units (neurons) are deactivated (set to zero). These hidden units produce outputs that represent the values generated by the hidden state and the input after

passing through both gates. The values can represent either the next layer's hidden state or the output of the stack, if applicable. Dropout is applied to the non-recurrent connections (i.e., the output of each GRU layer) but not to the recurrent connections (the hidden state passed between time steps).

2. **Dropout Layer:** Dropout [23] is applied between the stacked layers and the FC layer. This deactivates a subset the output generated from the stacked GRU, across the feature space before passing it into the FC layer. This regularization ensures that the model does not rely too heavily on any single feature, improving the models generalization since it learns to make predictions based on a broader set of features.

3. **FC Layer:** The FC [23] layer transforms hidden state vectors with a dimension of `hidden_size` to an output vector with a dimension of `number_of_states`. This output dimension matches the length of 3 in the one-hot encoded output, corresponding to the possible types: MA, EM, and BW.

4.4.1.3 Output Sequence

The final output of the model is a sequence of vectors with dimensions [`'batch_size'`, `'sequence_length'`, `'number_of_states'`]. Each vector at each time step contains raw logit scores for each noise type.

4.4.2 Model Process

4.4.2.1 Model Training

The training of the models is conducted over 200 epochs using early stopping criterion to prevent overfitting the model to the data. Each epoch consists of a forward pass, loss computation, backpropagation and model parameter updates.

1. **Forward Pass:** In each epoch, the model performs a forward pass by processing a batch of input data through the layers described in Section 4.4.1.2. During this process, each layer computes its output, which is then passed as input to the subsequent layer. The final output of the model has dimensions [`batch_size`, `sequence_length`, `number_of_states`], matching the target labels used for loss calculation.

2. **Loss Calculation:** The Binary Cross-Entropy with Logits Loss (BCEWithLogitsLoss) [23] function is used to measure the performance of the model in multi-label classification tasks. It computes the binary cross-entropy loss by combining the sigmoid activation function and the binary cross-entropy calculation into one numerically stable operation. This loss function takes the raw logits output from the forward pass, applies the sigmoid function to obtain probabilities, and then calculates how well these probabilities match the target labels. Additionally, BCEWithLogitsLoss

supports class weights to ensure that minority classes are adequately represented during training.

a) **Class Imbalance Problem:** In an imbalanced dataset, certain classes appear less frequently than others. For instance, MA and EM occur less often compared to other noise type. The class distribution is as follows: 16.0 % for EM, 16.0 % for MA, 27.4 % for BW, with the remaining 40.7 % representing instances with no noise. To address this, the loss function was adjusted by penalizing the misclassification of less frequent classes more heavily. This is done by assigning higher weights to these classes in the loss calculation. As a result, the model receives stronger feedback for errors involving the rare classes, which helps it to better learn and predict these underrepresented classes.

b) **Backward Pass and Optimization:** Gradients were computed using backpropagation, and the model parameters were updated with the Adaptive Moment Estimation (Adam) optimizer [23], set with a learning rate of 0.001. This optimizer was selected for its adaptive learning rate capabilities.

4.4.3 Validation Process

Validation is conducted after each epoch to assess the model's performance on unseen data and to guide the training process. Validation involves setting the model to evaluation mode, where gradient computation is disabled. This allows to objectively measure the model's effectiveness without influencing its learning.

1. **Validation Loss Calculation:** During validation, the loss is calculated using the same criterion as during training phase. However, unlike training, no backpropagation or parameter updates occur. The purpose of this step is to evaluate the model's ability to generalize to new data, which helps identify any potential overfitting to the training set.

2. **Evaluation and Model Saving:** The model's performance is evaluated based on the validation loss. If the validation loss decreases compared to previous epochs, it indicates that the model's ability to generalize has improved. In such cases, the model is then saved as the best version. This checkpoint includes the model's state, optimizer state, and all relevant hyperparameters, ensuring that the model could be restored.

3. **Early Stopping:** Early stopping is employed with a patience of 40 epochs to prevent overfitting. This means that if there is no improvement in validation loss for 40 consecutive epochs, the training process is terminated. Early stopping helps avoiding unnecessary training beyond the point of optimal performance, thereby reducing computational resources and ensuring that the model does not become overly specialized to the training data.

4.4.4 Hyperparameter Optimization Via Grid Search

To optimize the model's performance, hyperparameters were fine-tuned using a grid search approach. Grid search systematically explores a predefined set of hyperparameter values by training and validating the model on different combinations of these. This process assisted in identifying the optimal configuration that yields the best validation performance.

1. **Grid Search Methodology:** A grid search is performed by exhaustively evaluating every combination of the specified hyperparameters within a defined range, the overall tested values can be seen in Table 4.3. For each combination, the model is trained and validated, and the results were recorded. The combination that resulted in the lowest validation loss is selected as the optimal set of hyperparameters.

Table 4.3: Hyperparameter values explored during grid search

Hyperparameters	Values
Types of layers	GRU
Number of layers	3
Bidirectional	`True/False`
Batch size	128
Hidden size	64, 128, 256
Dropout rate	0, 0.3, 0.5

2. **Selection of the Best-Performing Set of Hyperparameters:** After conducting the grid search, the best-performing hyperparameter set is chosen based on its ability to minimize validation loss.

4.4.5 Testing Process

After training and validating the model, the final step is to evaluate its performance on the test dataset. This evaluation is crucial for assessing the model's ability to generalize to unseen data, allowing for the computation of key metrics such as confusion matrices, accuracy, F1 scores, precision, and recall.

Testing begins by loading the best model saved during training. This model is reinitialized with the same architecture and hyperparameters, and the model's state, including the learned weights, are restored.

At this stage, the model outputs raw logits for each time step are passed through a sigmoid function to produce probabilities. The probabilities are then converted into binary predictions, indicating the presence or absence of each noise type at each time step. This is achived using a class-specific threshold optimization in a multi-label classification setting to improve model performance. For each class, the Receiver Operating Characteristic (ROC) curve is computed, mapping the relationship between True Positives (TP) rate and False Positives (FP) rate at various thresholds. The G-Mean, or geometric mean

of TP rate FP rate, is calculated, as $\sqrt{\text{TPR} \times (1 - \text{FPR})}$ for each threshold to capture a balance between sensitivity and specificity. The threshold that maximizes the G-Mean is selected as the optimal value, ensuring that each class receives a threshold tailored to its individual distribution. The thresholds are calculated on validation set and then applied to the testing set, allowing an unbiased assessment of the model's generalization performance. This step adjusts the balance of the performance metrics. This method provides a refined evaluation approach that addresses the imbalanced nature of multi-label dataset by aligning thresholds with class-specific predictive behavior, resulting in more reliable performance metrics across all classes.

4.4.5.1 Performance Evaluation

To evaluate the model's performance, the predictions are compared to the actual labels. The following metrics are used for this assessment.

1. **Confusion Matrices**

   a) Confusion Matrix for each individual class:

   An individual confusion matrix is computed for each noise type to evaluate the model's ability to correctly predict whether each class (MA, EM, BW) is `'Present'` or `'Not Present'`. The matrix shows the counts of:

   - True Negatives (TN): Correctly predicted the noise type as `'Not Present'`
   - FP: Incorrectly predicted the noise type as `'Present'`
   - False Negatives (FN): Incorrectly predicted the noise type as `'Not Present'`
   - TP: Correctly predicted the noise type as `'Present'`

   b) Global Confusion Matrix: All Classes including a None class.

   A general encapsulating matrix is calculated to evaluate the model across four categories: MA [1, _, _], EM [_, 1, _], BW [_, _, 1], and None [0, 0, 0]. This matrix summarizes the frequency of noise misclassification and helps to identify which classes are most commonly confused. It is important to note that 'None' is not a distinct class but rather a result of no noise being present.

   - TP along the diagonal: Correct predictions where the true labels match the predicted labels.
   - Off-diagonal values: Misclassifications, where one class is incorrectly predicted as another.
   - None category: Indicates how well the model recognizes instances where none of the other classes are active.

2. **Performance Metrics**

   From the individual confusion matrices for each class, the following metrics are calculated to assess the model's performance:

- **Accuracy:** Accuracy measures the proportion of correct predictions out of the total number of data points. While it provides a general indication of the model's effectiveness, it can be misleading in cases of imbalanced data, as it may overestimate performance on the majority class. However, as mentioned in Section 4.4.2.1, penalizations were added to mitigate this effect and improve accuracy for the minority classes.

$$Accuracy : \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

- **Precision:** Precision indicates how many of the predicted positives were actually correct. It is particularly important when dealing with imbalanced data, as it reflects the model's ability to avoid false positives.

$$Precision : \frac{TP}{TP + FP} \tag{4.2}$$

- **Recall:** Recall, also known as sensitivity, measures the ability of the model to correctly identify actual positives. In the context of imbalanced data, recall is crucial for ensuring that the minority classes are accurately detected.

$$Recall : \frac{TP}{TP + FN} \tag{4.3}$$

- **F1 Score:** The F1 Score is the harmonic mean of precision and recall. It balances the trade-off between these two metrics.

$$F1Score : 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.4}$$

These evaluation metric provide a comprehensive and balanced understanding of the model's performance.

4.5 Classification and Output Interpretation

The output structure allows for easy translation into a readable format, enabling precise classification of the signal's noise contamination. It reports the overall percentage of noise, as well as the breakdown for each type (EM, MA, BW), and identifies the specific intervals where they occur. The final classification, as detailed in Figure 4.4, presents this information in a simplified format.

4.6 Tradional Methods Comparison

The method for comparing the effectiveness of traditional SQIs focuses on evaluating the performance of these metrics across the test set. This approach assumes that signals with added noise are noisy, regardless of the magnitude, and that the clean labeled

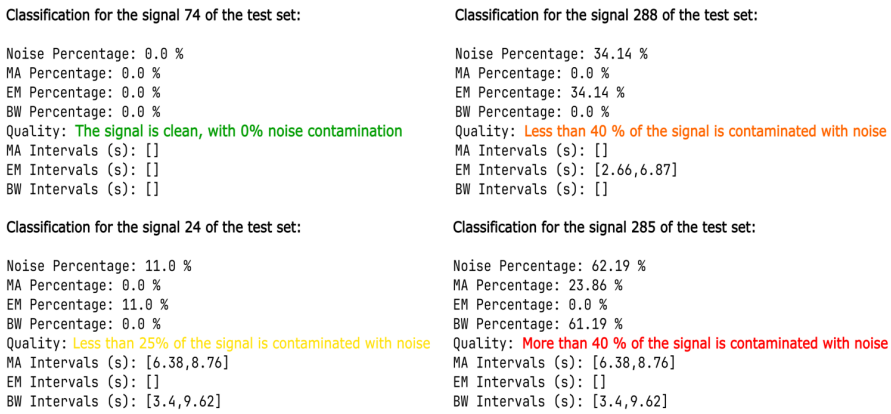


Figure 4.4: Examples of generated reports for the signals 74, 24, 288, 284 of the test set

signals are those prior to noise addition. To assess performance, the number of incorrect classifications by the SQIs, based on clean and noisy labels, is counted. Given the known thresholds for Kurtosis (kurt), Skewness (skew), Power Spectral Density (psd), and Baseline Relative Power (bas), the percentages of correct and incorrect classifications are calculated. For the Signal-to-Noise Ratio (SNR), both clean and noisy signal pairs are used. The thresholds used are presented in Table 4.4.

Table 4.4: Thresholds found in literature for clean ECG signals

Metric	Range	ref.
kurt	> 5	[11]
psd	> 0.9	[11]
bas	> 0.95	[11]
skew	$> -0.8 \cap \leq 0.8$	[10]
SNR	> 10 dB	[10]

4.7 Deep Learning Methods Comparison

The method used to compare the approach presented in this thesis with state-of-the-art studies involved simplifying the model's predictions. Instead of using a one-hot array for each time step, these were converted to a binary output per time step: one if there was any noise present and zero if no noise was detected. This allowed for the classification of signals into noisy or clean categories. After this conversion, performance metrics, as shown in Section 4.4.5.1, were calculated, allowing a closer comparison.

5

Results

This chapter presents the results of the study, focusing on the identification and classification of noise. The objective was to detect the presence of noise and classify its type to expand on a gap in the literature concerning the ECG SQI into acceptable and unacceptable classifications.

5.1 Optimal Architecture Via Grid Search

The optimal model configuration and hyperparameters were identified through a grid search. The final GRU-Based architecture, summarized in Table 5.1 and illustrated in Figure 4.3, consists of three GRU layers followed by one Dropout layer and FC layer, with a hidden size of 128, bidirectional setup, 0.3 dropout rate, 0.001 learning rate, and a batch size of 128. The model achieved its lowest validation loss of 0.34 at epoch 43. As shown in Figure 5.1, the training and validation loss curves demonstrate convergence to this optimal point.

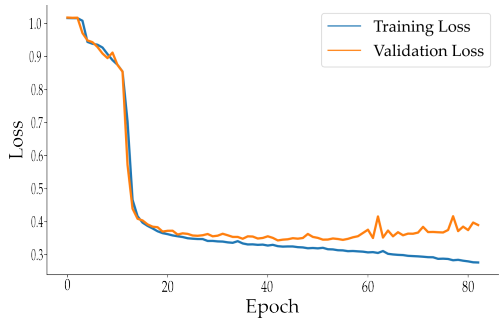


Figure 5.1: Loss curves for the training and validation processes

Hyperparameters	Values
Number of GRU layers (with internal Dropout)	3
Number of Dropout layers	1
Number of FC layers	1
Hidden Size	128
Bidirectional	True
Dropout Rate	0.3
Learning Rate	0.001
Batch Size	128
Best Validation Loss	**0.34**

Table 5.1: Model hyperparameters and best validation loss

5.2 Performance Metrics on the Test Set

The performance of the model was evaluated using accuracy, precision, recall, and F1-scores across three types of noise. The results are summarized in Table 5.2.

In terms of accuracy, the model performed best at identifying EM noise, achieving 92.86 %. This was followed by BW noise with 92.05 % accuracy, while the lowest accuracy was observed for MA noise at 81.55 %.

When analyzing precision, the model demonstrated its highest performance in detecting BW noise, with a precision of 82.36 %, indicating a lower false positive rate for this type of noise. The precision for EM and MA noise was lower, at 79.35 % and 50.37 %, respectively.

The recall metric, which measures the ability of the model to correctly identify the true positive cases, remained high across all noise types. BW noise achieved the highest recall at 96.56 %, followed by MA at 90.14 % with EM recording 85.26 %.

Finally, the F1 score, which balances precision and recall, further underscores the model's performance trends. BW noise achieved the highest F1 score at 88.89 %, followed by EM noise with 82.19 %, and MA noise with 64.62 %.

Table 5.2: Performance metrics

Metric	MA	EM	BW
Accuracy (%)	81.55	92.86	92.05
Precision (%)	50.37	79.35	82.36
Recall (%)	90.14	85.26	96.56
F1 Score (%)	64.62	82.19	88.89

5.3 Confusion Matrix Analysis

The confusion matrices for each type of noise, Figure 5.2, reveal distinct patterns in how the model distinguishes between the presence and absence of noise. It is important to note that the `'Not Present'` category has a higher count in these matrices, which is expected given that the signals consist mainly of clean recordings with intervals of noise.

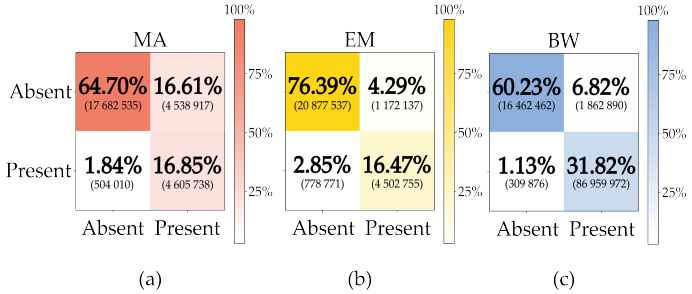


Figure 5.2: Matrices for each type of noise

For MA noise, Figure 5.2 (a) shows that 81.55 % of the instances were correctly classified regarding noise presence: 64.70 % were successfully identified as noise-free, while noise was correctly detected in 16.85 % of the cases. There were 1.84 % of cases where noise was incorrectly detected when it was actually absent, and 16.61 % of the instances where the noise was present but went undetected.

In the case of EM noise, Figure 5.2 (b) displays that 92.86 % of the cases were correctly classified regarding the presence or absence of noise: 76.39 % were accurately identified as having no noise, and 16.47 % were correctly recognized as containing noise. Misclassifications occurred in 2.85 % of the cases where noise was detected despite its absence. Furthermore, there were 4.29 % of the occurrences in which noise was present but not detected.

For BW noise in Figure 5.2 (c), 92.05 % of the instances were correctly classified regarding the presence of noise: 60.23 % were recognized as having no noise, and 31.82 % were identified as containing noise. False positives, where noise was incorrectly identified, accounted for 1.13 % of the cases. False negatives, where noise was present but not detected, made up 6.82 % of all occurrences.

The overview matrix in Figure 5.3 provides a structured evaluation of the model's performance across different noise categories and clean intervals. The None category, which represents periods without noise, shows a high proportion of correctly identified instances, 99.73 %. Misclassifications in this category are minimal, with 0.12 % incorrectly identified as MA, 0.08 % as EM, and 0.08 % as BW. As mentioned previously, the None category will accumulate more occurrences due to the structure of the input signals.

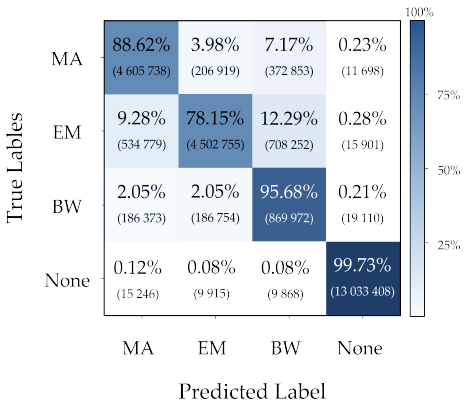


Figure 5.3: Overview matrix

For BW, the model achieves a correct classification rate of 95.68 %. Some confusion occurs between BW and other noise types, with 2.05 % of BW instances misclassified as EM and 2.05 % as MA.

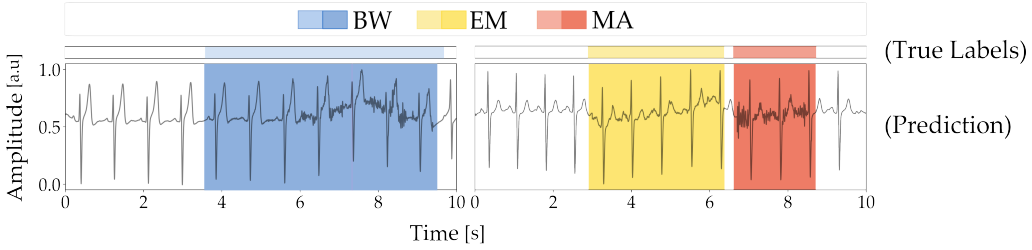
EM shows a correct classification in 78.15 % of instances, misclassifications are more prominent in this category, with 12.29 % of EM instances being misidentified as BW and 9.28 % as MA.

MA has a correct classification rate of 88.62 %, misclassification rates in this category

are higher, with 7.17 % of MA instances confused with BW and 3.98 % with BW.

5.4 Visualization of Model Predictions

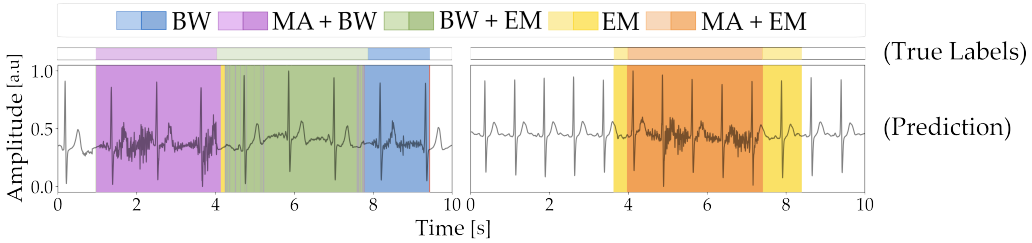
Figure 5.4 presents two signals from the test set, demonstrating the model's ability to classify isolated noise types without overlap. In Signal 265 (a), the model accurately detects BW noise, matching the true labels. In Signal 1482 (b), it effectively distinguishes between distinct segments of EM and MA noise. These examples showcase the model's high accuracy in identifying various noise types when they occur independently.



(a) Signal 265 from the test set (b) Signal 1482 from the test set

Figure 5.4: Model's prediction with accurate classification of the different noise types isolated

Figure 5.5 illustrates the model's performance with two overlapping noise types. In Signal 6082 (a), the model correctly identifies combinations of BW with MA and BW with EM. Signal 5316 (b) also shows the accurate classification of overlapping segments of MA and EM.

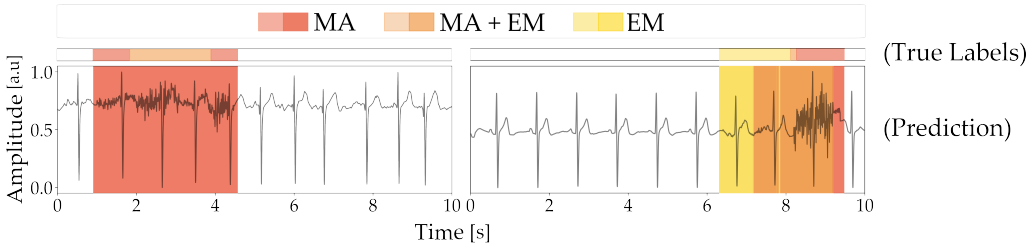


(a) Signal 6082 from the test (b) Signal 5316 from the test

Figure 5.5: Model's prediction with accurate classification of two overlapping noise types

In contrast, Figure 5.6 in Signal 1181 (a), it can detect the noisy segments but misclassifies EM as MA. In Signal 898 (b), it correctly identifies the presence of noise but missclassifies MA with EM.

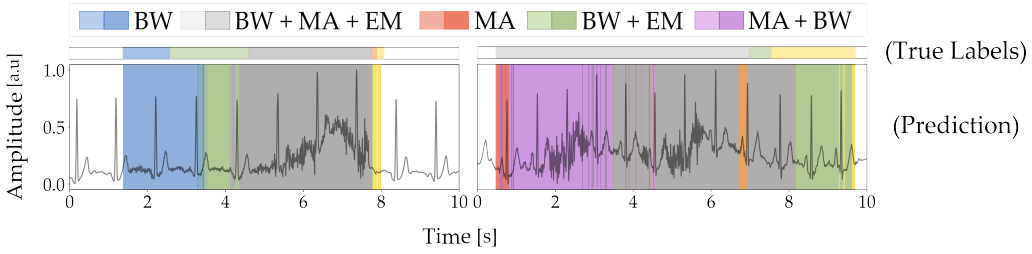
Figure 5.7 demonstrates the model's performance in scenarios involving three overlapping noise types. In Signal 5987 (a), the model accurately classifies the complex combination of BW, MA, and EM noise. However, in Signal 697 (b), the model fails to differentiate the noise types correctly, resulting in misclassifications.



(a) Signal 1181 from the test set with EM noise is classified as MA

(b) Signal 898 from the test set with MA noise is classified as EM

Figure 5.6: Model's prediction with correct detection of noisy segments with incorrect classification of their types



(a) Accurate prediction of the three types of noise combined (signal 5987 from the test set)

(b) Innacurate prediction of the three types of noise combined (signal 697 from the test)

Figure 5.7: Model's prediction with classification of the three overlapping noise types

5.5 Traditional Metrics as an Assessment Tool

As shown in Table 5.3, the SNR achieved the highest correct classification rate with 88.70 %, followed by 74.54 % skew, kurt at 54.12 % and bas at 53.17 %. A lower correct classification rates were observed for psd at 52.83 %.

Table 5.3: Performance of traditional metrics in the test set

Metric	Signal Type	Range	Min	Max	Mean	SD	Incorrect %	Correct %
kurt [11]	Noisy	≤ 5	-0.812	39.445	9.780	6.013	45.88	54.12
	Clean	> 5	-1.150	61.227	11.917	6.772		
psd [11]	Noisy	≤ 0.9	0.192	0.995	0.719	0.110	47.17	52.83
	Clean	> 0.9	0.364	0.998	0.756	0.109		
bas [11]	Noisy	≤ 0.95	0.598	1.000	0.983	0.026	46.83	53.17
	Clean	> 0.95	0.902	1.000	0.997	0.005		
skew [10]	Noisy	$\leq -0.8 \cup > 0.8$	-5.105	5.809	0.344	2.407	32.41	74.54
	Clean	$> -0.8 \cap \leq 0.8$	-5.438	5.971	0.409	2.762		
SNR [10]	Noisy and Clean	clean if 10 dB	-4.138	48.682	20.577	8.289	11.30	88.70

5.6 Binary Classification: Performance Evaluation

This section evaluates the adaptations described in Section 4.7, which enable binary classification of signals as either clean or noisy. These adaptations facilitate a direct comparison with deep learning methods that exclusively classify signals in this manner.

An accuracy of 99.72 %, precision of 99.78 %, recall of 99.68 %, and F1 score of 99.73 % were achieved.

6

Discussion

This chapter presents a comprehensive discussion organized into several key sections. First, attention focuses on the final model architecture, followed by an analysis of classification results to examine potential patterns and explore the underlying reasons for their occurrence. Next, a comparison with state-of-the-art methods contextualizes the model's performance within the field. Finally, the discussion addresses the model's strengths and limitations, as well as potential avenues for future work.

6.1 Training Process

The final architecture of the model, illustrated in Figure 4.3, was achieved through an extensive grid search method, where the best performing model was found from multiple configurations. This iterative process led to the selection of a model that demonstrated the lowest validation loss of 0.34, achieved at epoch 43.

The behavior of the training and validation loss values throughout the training process is depicted in Figure 5.1. Here, it is possible to observe the trends in both training and validation losses, the curves provide insight into the model's learning process and the effectiveness of the chosen architecture. Initially, a steep decline occurs in both training and validation losses, primarily due to the random initialization of weights. The slope of the curve decreases rapidly, particularly evident from epoch 0 to 20, indicating that the model is rapidly learning and adjusting its weights to capture underlying patterns in the data towards the minimization of the loss function. As training progresses, the validation loss exhibits noticeable fluctuations and occasional spikes. These variations can be attributed to weight adjustments during training that do not lead to improvements in the validation set, as well as to the presence of dropout, which introduces variability during the training process. Initially, the spikes in the validation loss curve are prominent, but they tend to reduce as training continues, leading the curve to flatten and approach the training loss curve, illustrating that the model is reaching a point of diminishing returns in terms of learning. The point in which learning ceases to yield improvements is associated with the lowest validation loss observed, achieved at epoch 43, just before its

subsequent increase. This is evidenced by the widening of the gap between the training and validation loss curves. Continuing to train beyond this point would compromise the model's generalization ability, potentially leading to overfitting. The implemented early stopping mechanism effectively halts the training process before this decline occurs.

6.2 Performance Metrics on the Test Set

The performance metrics in conjunction with the individual confusion matrices, provide valuable information into the model's ability to correctly classify each noise type, identifying instances of false positives, false negatives and overall accuracy for each class. While accuracy offers an overview of performance, it may not fully capture the impact of errors due to the imbalanced dataset. Therefore, alongside accuracy, particular attention will be given to precision, recall and the class matrices, as these help identify whether a class is being falsely detected, associated with FP, or overlooked when present, linked to FN instances.

The class that performed overall the best in all metrics detailed in Section 4.4.5.1 was the BW, achieving an accuracy of 92.05 %, correctly classifying 60.23 % of instances as BW-free and 31.82 % as having the noise present. The low rate of FP (1.13 %) and FN (6.82 %) aligns with the high precision and recall rates, confirming that the model is effective at recognizing BW when present and absent.

Class EM exhibits the highest accuracy among the classes, reaching a accuracy of 92.86 %, accurately classifying 76.39 % of the instances as free of EM and 16.47 % as containing the class. The rate of FP is now higher than anticipated at 4.29 %, while the FN rate is 2.85 %. This leads to decreased precision and recall scores of 79.35 % and 85.26 %, respectively, with the precision being the lowest among the three categories. These shifts indicate that the model is detecting the class more frequently than expected, and occasionally overlooking it.

The lowest accuracy is observed in class MA (81.55 %), identifying 64.70 % of the instances as not containing the class and 16.85 % as present with MA. In this case, the rates of FP and FN align with the trends observed for the EM class, but with a higher FP rate of 16.61 % and a lower FN rate of 1.84 %. The elevated FP rate contributes to the lowest precision among the three classes at 50.37 %, while the recall stands at 90.14 %. These metrics indicate that this class is being detected more often than expected and, to a lesser extent, not being accurately predicted when present.

The F1 score provides a balanced measure by combining precision and recall. The highest F1 score for BW noise at 88.89 % reinforces the notion that the model identifies this noise type correctly, maintaining a good balance between precision and recall. In contrast, the F1 scores for EM (82.19 %) and MA (64.62 %) noises are lower, reflecting the model's challenges in managing both FP and FN effectively.

It is important to note that despite some decrease in recall values between classes, the range achieved of 85.26 % to 96.56 % indicates that the model effectively identifies positive

instances. This suggests that the use of class weights, as discussed in Section 4.4.2.1, has improved the model's performance, even in the presence of a disproportionate number of negative instances (noise-free instances).

6.3 Comparative Evaluation of the Classes

To comprehend the reasons behind the accurate detection of BW and the lower F1 scores for the other classes (MA and EM), as well as the excessive detection of EM when it is absent and the oversight of EM when present, it is essential to examine the overview confusion matrix, displayed in Figure 5.3.

In cases where different noise types are overlapped in the same segment of signal, a factor that is likely contributing to the occurrence of misclassifications is the scale factor that is applied to each noise type before adding it to the interval. Since each noise is multiplied by a random scale factor, some types may be more prominently represented in the interval. This uneven representation can result in predictions where the model classifies the dominant type, leading to misclassification of less pronounced noises, this is evident in case of overlapping noise, as shown in the Figures 5.6 and 5.7.

Given this context, the discussion will focus related misclassification patterns and possible causes. The BW is identified correctly 95.68 % of all instances, this is expected since it is a low-frequency noise (typically around 0.5 to 1 Hz) [15]. Shows some minor misclassification as EM (2.05 %) and MA (2.05 %).

Meanwhile the MA class was properly detected with a rate of 88.62 %, with more significant misclassifications, note that this class is occasionally overlooked when present so it is being misclassified as BW 7.17 % of the times and as EM, less frequently, sitting at a 3.98 % rate.

The class EM has the lowest percentage among the classes with a value of 78.15 %, the misidentifications being more prominent, as BW has a rating of 12.29 % and as MA a value of 9.28 %. The EM has lowest recall, which is confirmed by the rates of misidentification, indicating the high number of FN.

The incorrect categorization of classes as BW is probably the result of the overrepresentation of BW, which occurs due to its minimum duration of 5 seconds each time it is present, leading to increased representation in the dataset. Although the recall values for all classes are relatively high, indicating that the impact may not be significant, a closer examination of the overview confusion matrix reveals noticeable effects. The misclassification of EM as MA highlights the higher number of FP associated with MA and the increased FN for EM. Regardless, the opposite still occurs, as seen in Figure 5.6, where the MA is being identified as EM, this aligns with the results provided by confusion matrix, recall and precision scores. This may result from the characteristics of both noise types, which can have overlapping frequency bands. MA typically ranges from 0.01 to 100 Hz [16], and EM, which falls between 1 and 10 Hz [14]. These similarities can make it challenging for the model to distinguish between their features.

Overall, the model demonstrates several positive aspects, showing a strong potential to detect and classify noise in ECGs and its effectiveness in recognizing these noise types, including some combinations of overlapping noise, although it is not always successful in more complex situations.

6.4 Comparison with Traditional Metrics

Although a direct comparison between traditional metrics and DL methods is not possible due to differences in approach, it is worth noting the contrast in performance contrast in performance when doing a simple binary test of distinguishing signals containing segments with noise versus noise-free signals (ie, noisy versus clean signals). In this experiment, traditional metrics like SNR achieved 88.70 % accuracy, while the majority of others fell below 75 %, as shown in Table 5.3. However, SNR's practical utility is limited, as it requires access to both clean and noisy versions of a signal, a requirement rarely met in real-world settings where clean signals are typically unavailable. In contrast, the model presented here exhibits significantly higher performance, highlighting the potential advantages of DL approaches for SQA, this can be illustrated by the Annex I. This disparity further motivates the need for exploring DL techniques in this domain.

6.5 Comparison with other Deep Learning Methods

The model excels at identifying binarily whether a signal contains noise and pinpointing its location, regardless of its magnitude, achieving an accuracy of 99.72 %, precision of 99.78 %, recall of 99.68 %, and F1 score of 99.73 %. In comparison, the highest accuracy reported in Table 3.1 is 97.72 % [49]. The cited work defines a signal as noisy if it disrupts the readability of the QRS complex, which hinders the ability to make a direct comparison. Furthermore, it is important to note that an objective comparison is challenging, as many referenced papers do not provide all the metrics used in this thesis. The model also effectively extends this approach by accurately identifying which segments are noisy and classifying its type.

6.6 Limitations and Future Work

Despite its strengths, the model exhibits certain limitations that require consideration. Its performance declines when handling overlapping noise types, particularly in distinguishing between EM and MA noises. While the results are satisfactory, there is potential for further improvements. One approach to address this limitation involves integrating an attention mechanism. By incorporating an attention layer after the stacked GRU layers, that allows the model to focus on the characteristic features of each noise type, allowing for better differentiation between overlapping segments and improve classification results.

In a clinical context, this model could be expanded to receive input from the 12-lead ECG, enabling the identification of specific types of noise and providing real-time feedback for medical professionals to take proper measures during recordings. Meanwhile ambulatory settings, the model could similarly alert patients to issues and offer guidance for corrections.

Additionally, this model contributes to research by expanding current DL approaches, offering a tool that not only detects noise but also categorizes it. This capability can make denoising methods more efficient, both traditional and DL-based, by highlighting the noisy segments that would be subject to cleaning and directing the appropriate denoising mechanisms to the specific type of noise present. Another potential application is the creation of automatically labeled databases, particularly since most research datasets stem from wearable-acquired data.

A key point to consider is that the model currently focuses on temporal quantization rather than noise intensity. To improve its usability in the mentioned contexts, incorporating noise level values into the temporal quantization would be beneficial. Currently, the model outputs a one-hot array for each timestep to identify active noise types. This approach could be enhanced by applying transfer learning to include noise intensity alongside noise type. By leveraging existing temporal features, the output vector could be extended to capture continuous values representing noise levels, quantifying the difference between the clean and noisy signals.

The model achieves good results, and has a relatively simple architecture, which contributes to its computational efficiency. However, it currently employs a sample-to-sample approach, produces very long outputs without the need for that, since a sample-wise detection of noise is not necessary. For example, analyzing a 10-minute signal at 360 Hz would produce an output of 216,000 samples, making it impractical for real-time applications. A more efficient solution could involve adopting an interval-based approach, where the model summarizes noise detection over fixed time intervals, such as 1-second windows, reducing the output to 360 samples. This would significantly decrease the output size while maintaining adequate accuracy, making it more suitable for integration into real-time, user-friendly tools.

Another consideration is the usage of real datasets to test and train the model. The training could be fine-tuning the model by training some epochs with a real dataset to seek improvements. Testing would require extensive manual labeling, and real-world signals often lacks clearly distinguishable noise types, making this a challenging adjustment.

7 Conclusion

The primary objective of this thesis is the development and evaluation of a DL model capable of performing SQA by detecting and classifying noisy segments in ECG. To achieve this, a labeled, controlled noisy ECG dataset was generated, and a GRU-based model was designed to identify and categorize noise in time series data.

The results indicate that the model excels in detecting the different noise types achieving an accuracy of 92.86 % for EM and 92.05 % , 81.55 % for BW and MA. Misclassifications are likely due to the morphological and frequency similarities of these noise types. In cases of overlapping noise, the chances of occurring misclassifications are higher, due to increased difficulty in distinguishing the different classes and one class being more dominant than the other(s). Despite these challenges, the model maintained high recall and precision rates, demonstrating its effectiveness in identifying noise when present, with the exception of the precision of MA, due to the higher FP rates. This thesis supports the evidence that DL may be more effective than traditional methods for SQA. While conventional methods like SNR achieve only 88.70 % accuracy or lower, the proposed model significantly outperforms them. Moreover, when comparing to other DL approaches, the model achieves 99.72 % accuracy for binary classification, surpassing the highest results reported in the current literature. Nevertheless, there are limitations to be addressed. The model encounters challenges with overlapping noise types, especially between EM and MA. This could be improved by incorporating attention mechanisms to enhance the model's focus on distinctive noise features. The current model detects noise presence but not intensity. Adding noise level quantification via transfer learning could expand its use. Shifting from a sample-based to an interval-based method may improve efficiency for real-time applications in ambulatory settings. A challenge remains in testing with real-world datasets due to the difficulty of manually labeling noise.

By refining noise classification beyond clean versus noisy, this work improves the model's reliability in continuous monitoring systems, benefiting clinicians, researchers, wearable device users, and individuals with interest in signal processing while creating opportunities for clinical and research advancements.

In conclusion, the objectives outlined in the Introduction are achieved, namely the

development of a DL model capable of identifying noise types in ECG signals, an evaluation of the model's performance in binary classification, and a comparison with state-of-the-art methods.

Bibliography

[1] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/main/template.pdf (cit. on p. i).

[2] W. H. Organization. *Cardiovascular diseases*. https://www.who.int/health-topics/cardiovascular-diseases. Accessed: 2024/09/24 (cit. on p. 1).

[3] R. Kher. "Signal Processing Techniques for Removing Noise from ECG Signals". In: *Journal of Biomedical Engineering and Research* 1 (2019), pp. 1–9. URL: https://www.jscholaronline.org/full-text/JBER/3_101/Signal-Processing.php (cit. on p. 1).

[4] M. Carrington et al. "Monitoring and diagnosis of intermittent arrhythmias: evidence-based guidance and role of novel monitoring strategies". In: *Eur Heart J Open* 2.6 (2022). DOI: 10.1093/ehjopen/oeac072 (cit. on p. 1).

[5] A. H. Association. *Holter Monitor*. https://www.heart.org/en/health-topics/heart-attack/diagnosing-a-heart-attack/holter-monitor. Accessed: 2024/09/24 (cit. on pp. 1, 4).

[6] S.-H. Liu et al. "Development of a Patch-Type Electrocardiographic Monitor for Real Time Heartbeat Detection and Heart Rate Variability Analysis". In: *Journal of Medical and Biological Engineering* 38.6 (2018), pp. 411–423. DOI: 10.1007/s40846-018-0369-y (cit. on pp. 1, 4).

[7] A. Pingitore et al. "An overview of the electrocardiographic monitoring devices in sports cardiology: Between present and future". In: *Clinical Cardiology* 46.9 (2023), pp. 1028–1037. DOI: 10.1002/clc.24073. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/clc.24073 (cit. on p. 1).

[8] A. Baldassarre et al. "The Role of Electrocardiography in Occupational Medicine, from Einthoven's Invention to the Digital Era of Wearable Devices". In: *Int J Environ Res Public Health* 17.14 (2020-07), p. 4975. DOI: 10.3390/ijerph17144975 (cit. on p. 1).

[9] E. S. Dahiya et al. "Wearable Technology for Monitoring Electrocardiograms (ECGs) in Adults: A Scoping Review". In: *Sensors* 24.4 (2024). ISSN: 1424-8220. DOI: 10.3390/s24041318. URL: https://www.mdpi.com/1424-8220/24/4/1318 (cit. on p. 1).

[10] M. S. Rahman et al. "Robustness of electrocardiogram signal quality indices". In: *Journal of The Royal Society Interface* 19 (2022-04). DOI: 10.1098/rsif.2022.0012 (cit. on pp. 2, 10, 27, 32, 48).

[11] Z. Zhao and Y. Zhang. "SQI Quality Evaluation Mechanism of Single-Lead ECG Signal Based on Simple Heuristic Fusion and Fuzzy Comprehensive Evaluation". In: *Frontiers in Physiology* 9 (2018-06). DOI: 10.3389/fphys.2018.00727 (cit. on pp. 2, 10, 27, 32, 48).

[12] K. van der Bijl, M. Elgendi, and C. Menon. "Automatic ECG Quality Assessment Techniques: A Systematic Review". In: *Diagnostics* 12.11 (2022). Ed. by A. El-Baz, p. 2578. DOI: 10.3390/diagnostics12112578 (cit. on p. 2).

[13] P. Woodrow. "Introduction to electrocardiogram interpretation: part 2." In: *Emergency nurse : the journal of the RCN Accident and Emergency Nursing Association* 18 (2 2010-05), pp. 28–36. ISSN: 1354-5752. URL: http://www.ncbi.nlm.nih.gov/pubmed/20527455 (cit. on p. 4).

[14] M. Sugadev et al. "Survey on Various noise sources in ECG signal and its Filtering Methods". In: *2021 10th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON)* (2021), pp. 01–06. DOI: 10.1109/IEMECON53809.2021.9689132 (cit. on pp. 5, 36).

[15] A. C. V. Maggio et al. "Quantification of Ventricular Repolarization Dispersion Using Digital Processing of the Surface ECG". In: *Advances in Electrocardiograms - Methods and Analysis* (2012-01), pp. 183–185. DOI: 10.5772/23050. URL: http://www.intechopen.com/books/advances-in-electrocardiograms-methods-and-analysis/quantification-of-cardiac-ventricular-repolarization-dispersion-using-computarized-ecg- (cit. on pp. 5, 36).

[16] S. Chatterjee et al. "Review of noise removal techniques in ECG signals". In: *IET Signal Processing* (2020). DOI: 10.1049/iet-spr.2020.0104 (cit. on pp. 5, 36).

[17] U. Satija, B. Ramkumar, and M. S. Manikandan. "A Review of Signal Processing Techniques for Electrocardiogram Signal Quality Assessment". In: *IEEE Reviews in Biomedical Engineering* 11 (2018), pp. 36–52. ISSN: 1937-3333. DOI: 10.1109/RBME.2018.2810957 (cit. on p. 5).

[18] Z.-H. Zhou. *Machine Learning*. Springer Singapore, 2021. ISBN: 978-981-15-1966-6. DOI: 10.1007/978-981-15-1967-3. URL: https://link.springer.com/10.1007/978-981-15-1967-3 (cit. on p. 5).

[19] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning". In: *Nature* 521 (7553 2015-05), pp. 436–444. ISSN: 0028-0836. DOI: 10.1038/nature14539. URL: https://www.nature.com/articles/nature14539 (cit. on p. 6).

[20] W. Zhang et al. "On Definition of Deep Learning". In: *2018 World Automation Congress (WAC)* (2018-06), pp. 1–5. DOI: 10.23919/WAC.2018.8430387. URL: https://ieeexplore.ieee.org/document/8430387/ (cit. on p. 6).

[21] R. Qamar and B. A. Zardari. "Artificial Neural Networks: An Overview". In: *Mesopotamian Journal of Computer Science* (2023-08), pp. 124–133. ISSN: 0019462X. DOI: 10.58496/MJCSC/2023/015. URL: https://mesopotamian.press/journals/index.php/cs/article/view/118 (cit. on p. 6).

[22] R. Kashimpure. "Artificial neural networks (ANNs, also shortened to neural networks (NNs) or neural nets)". In: *International Research Journal of Modernization in Engineering Technology Science* (2023-07). ISSN: 25825208. DOI: 10.56726/IRJMETS43049. URL: https://www.irjmets.com/uploadedfiles/paper//issue_7_july_2023/43049/final/fin_irjmets1689177621.pdf (cit. on p. 6).

[23] A. Jha. *Mastering PyTorch - Second Edition: Create and Deploy Deep Learning Models from CNNs to Multimodal Models, LLMs, and Beyond*. Expert insight. Packt Publishing, 2024. ISBN: 9781801074308. URL: https://books.google.pt/books?id=eoZ7zwEACAAJ (cit. on pp. 6–9, 22, 23).

[24] S. Wang, T. Zhou, and J. Bilmes. "Bias Also Matters: Bias Attribution for Deep Neural Network Explanation". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019-09–15 Jun, pp. 6659–6667. URL: https://proceedings.mlr.press/v97/wang19p.html (cit. on p. 6).

[25] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. URL: http://www.deeplearningbook.org (cit. on p. 7).

[26] R. C. Staudemeyer and E. R. Morris. "Understanding LSTM - a tutorial into Long Short-Term Memory Recurrent Neural Networks". In: *ArXiv* abs/1909.09586 (2019). URL: https://api.semanticscholar.org/CorpusID:268130517 (cit. on p. 8).

[27] F. M. Bianchi et al. "Recurrent Neural Network Architectures". In: *Recurrent Neural Networks for Short-Term Load Forecasting: An Overview and Comparative Analysis*. Cham: Springer International Publishing, 2017, pp. 23–29. ISBN: 978-3-319-70338-1. DOI: 10.1007/978-3-319-70338-1_3 (cit. on p. 8).

[28] S. Obeta, E. Grisan, and C. V. Kalu. "A Comparative Study of Long Short-Term Memory and Gated Recurrent Unit". In: *SSRN Electronic Journal* (2023). ISSN: 1556-5068. DOI: 10.2139/ssrn.4442677 (cit. on p. 9).

[29] S. Yang, X. Yu, and Y. Zhou. "LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example". In: *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*. 2020, pp. 98–101. DOI: 10.1109/IWECAI50956.2020.00027 (cit. on p. 9).

[30] S. Oh. "A New Quality Measure In Electrocardiogram Signal". Mater thesis. University of Florida, 2004-12. URL: https://ufdc.ufl.edu/ufe0007281/00001 (cit. on pp. 10, 11).

[31] B. Rio, T. Lopetegi, and I. Romero. "Assessment of different methods to estimate electrocardiogram signal quality". In: *Computing in Cardiology* 38 (2011-01), pp. 609–612 (cit. on pp. 10, 11).

[32] F. Chiarugi et al. "Adaptive threshold QRS detector with best channel selection based on a noise rating system". In: *Computers in Cardiology*. 2007, pp. 157–160. DOI: 10.1109/CIC.2007.4745445 (cit. on p. 11).

[33] L. Li. "A Quality Assessment Method of Single-Lead ECG Signal Based on Spectral Analysis". In: *8th International Conference on Information Technology in Medicine and Education (ITME)* (2016-12), pp. 35–38. DOI: 10.1109/ITME.2016.0018 (cit. on p. 11).

[34] J. Wang. "A new method for evaluating ECG signal quality for multi-lead arrhythmia analysis". In: *Computers in Cardiology* 29 (2002-10), pp. 85–88. DOI: 10.1109/CIC.2002.1166713 (cit. on p. 11).

[35] S. Iravanian and L. Tung. "A novel algorithm for cardiac biosignal filtering based on filtered residue method". In: *IEEE Transactions on Biomedical Engineering* 49.11 (2002-11), pp. 1310–1317. ISSN: 0018-9294. DOI: 10.1109/TBME.2002.804589 (cit. on p. 11).

[36] X. Zhou et al. "ECG Quality Assessment Using 1D-Convolutional Neural Network". In: *14th IEEE International Conference on Signal Processing (ICSP)*. 2018-08, pp. 780–784. DOI: 10.1109/ICSP.2018.8652479 (cit. on pp. 12, 15).

[37] I. Silva, G. Moody, and L. Celi. "Improving the Quality of ECGs Collected Using Mobile Phones: The PhysioNet/Computing in Cardiology Challenge 2011". In: *Computing in Cardiology*. Vol. 38. 2011-01, 273276 (cit. on pp. 12, 15).

[38] G. D. Clifford et al. "AF classification from a short single lead ECG recording: The PhysioNet/computing in cardiology challenge 2017". In: *2017 Computing in Cardiology (CinC)*. IEEE. 2017, pp. 1–4. DOI: 10.22489/CinC.2017.065-469 (cit. on pp. 12, 15).

[39] A. L. Goldberger et al. "PhysioBank, PhysioToolkit, and PhysioNet". In: *Circulation* 101 (23 2000-06). ISSN: 0009-7322. DOI: 10.1161/01.CIR.101.23.e215 (cit. on pp. 12, 15, 16).

[40] A. Huerta et al. "Quality Assessment of Very Long-Term ECG Recordings Using a Convolutional Neural Network". In: *2019 E-Health and Bioengineering Conference (EHB)*. 2019-11, pp. 1–4. DOI: 10.1109/EHB47216.2019.8970077 (cit. on pp. 12, 15).

[41] Á. Huerta et al. "Comparison of Pre-Trained Deep Learning Algorithms for Quality Assessment of Electrocardiographic Recordings". In: *2020 International Conference on e-Health and Bioengineering (EHB)*. 2020, pp. 1–4. DOI: 10.1109/EHB50910.2020.9280217 (cit. on pp. 12, 15).

[42] A. Mondal, M. S. Manikandan, and R. B. Pachori. "Convolutional Neural Network Based ECG Quality Assessment Using Derivative Signal". In: *2022 Fourth International Conference on Cognitive Computing and Information Processing (CCIP)*. 2022-12, pp. 1–5. DOI: 10.1109/CCIP57447.2022.10058688 (cit. on pp. 12, 15).

[43] G. Liu et al. "ECG quality assessment based on hand-crafted statistics and deep-learned S-transform spectrogram features". In: *Computer Methods and Programs in Biomedicine* 208 (2021), p. 106269. ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2021.106269 (cit. on pp. 12, 15).

[44] J. Zhang et al. "A Signal Quality Assessment Method for Electrocardiography Acquired by Mobile Device". In: *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2018, pp. 1–3. DOI: 10.1109/BIBM.2018.8621160 (cit. on pp. 13, 15).

[45] X. Zhou et al. "Electrocardiogram Quality Assessment with a Generalized Deep Learning Model Assisted by Conditional Generative Adversarial Networks". In: *Life (Basel)* 11.10 (2021), p. 1013. ISSN: 2075-1729. DOI: 10.3390/life11101013 (cit. on pp. 13, 15).

[46] Y. Jin et al. "A novel attentional deep neural network-based assessment method for ECG quality". In: *Biomedical Signal Processing and Control* 79 (2023-12), p. 104064. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2022.104064. URL: https://www.sciencedirect.com/science/article/pii/S1746809422005341 (cit. on pp. 13, 15).

[47] M. Zhong et al. "Quality Assessment of Electrocardiogram Signals Using Contrastive Learning". In: *2023 13th International Conference on Information Technology in Medicine and Education (ITME)*. 2023, pp. 323–328. DOI: 10.1109/ITME60234.2023.00073 (cit. on pp. 13, 15).

[48] G. Chen et al. "SwinDAE: Electrocardiogram Quality Assessment Using 1D Swin Transformer and Denoising AutoEncoder". In: *IEEE Journal of Biomedical and Health Informatics* 27.12 (2023), pp. 5779–5790. DOI: 10.1109/JBHI.2023.3314698 (cit. on pp. 13, 15).

[49] X. Zhang et al. "Deep Learning-Based Signal Quality Assessment for Wearable ECGs". In: *IEEE Instrumentation Measurement Magazine* 25.5 (2022-08), pp. 41–52. DOI: 10.1109/MIM.2022.9832823 (cit. on pp. 13, 15, 37).

[50] Z. Cai et al. "An Open-Access Long-Term Wearable ECG Database for Premature Ventricular Contractions and Supraventricular Premature Beat Detection". In: *Journal of Medical Imaging and Health Informatics* 10.11 (2020), pp. 2663–2667. DOI: 10.1166/jmihi.2020.3289 (cit. on pp. 13, 15).

[51] H. Khamis et al. *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*. Version V3. 2016. DOI: 10.7910/DVN/QTG0EP (cit. on p. 15).

[52] G. Moody and R. Mark. "The impact of the MIT-BIH Arrhythmia Database". In: *IEEE Engineering in Medicine and Biology Magazine* 20.3 (2001), pp. 45–50. DOI: 10.1109/51.932724 (cit. on p. 15).

[53] PhysioNet. *MIT-BIH Normal Sinus Rhythm Database*. Available online: https://physionet.org/content/nsrdb/1.0.0/. 2000 (cit. on p. 15).

[54] R. Bousseljot, D. Kreiseler, and A. Schnabel. "Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet". In: *Biomedizinische Technik / Biomedical Engineering* 40.s1 (1995), pp. 317–318. DOI: 10.1515/bmte.1995.40.s1.317 (cit. on p. 15).

[55] W. Patrick et al. *PTB-XL, a large publicly available electrocardiography dataset (version 1.0.3)*. 2022. DOI: 10.13026/kfzx-aw45. URL: https://physionet.org/content/ptb-xl/1.0.3/ (cit. on pp. 15–17).

[56] A. Nemcova et al. *Brno University of Technology ECG Quality Database (BUT QDB) (version 1.0.0)*. 2020. DOI: 10.13026/kah4-0w24 (cit. on p. 15).

[57] G. Van Rossum and F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697. URL: https://www.python.org (cit. on p. 16).

[58] J. Ansel et al. "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation". In: *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, 2024-04. DOI: 10.1145/3620665.3640366. URL: https://pytorch.org/assets/pytorch2-2.pdf (cit. on p. 16).

[59] NVIDIA Corporation. *NVIDIA RTX 6000 Ada Generation Graphics Card*. https://www.nvidia.com/en-us/design-visualization/rtx-6000/. 2024 (cit. on p. 16).

[60] The pandas development team. *pandas-dev/pandas: Pandas*. Version v2.2.2. 2024-04. DOI: 10.5281/zenodo.10957263. URL: https://pandas.pydata.org (cit. on p. 16).

[61] C. R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (2020-09), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://numpy.org (cit. on p. 16).

[62] C. Xie et al. *Waveform Database Software Package (WFDB) for Python*. Version 4.1.0. 2023-01. DOI: 10.13026/9njx-6322. URL: https://physionet.org/content/wfdb-python/4.1.0/ (cit. on p. 16).

[63] The Matplotlib Development Team. *Matplotlib: Visualization with Python*. Version v3.9.2. 2024-08. DOI: 10.5281/zenodo.13308876. URL: https://matplotlib.org (cit. on p. 16).

[64] JetBrains. *PyCharm: The Python IDE for data science and web development*. Version 2024.1.4. 2024-08. URL: https://www.jetbrains.com/pycharm/ (cit. on p. 16).

[65] P. Wagner et al. "PTB-XL, a large publicly available electrocardiography dataset". In: *Scientific Data* 7 (1 2020-05), p. 154. ISSN: 2052-4463. DOI: 10.1038/s41597-020-0495-6 (cit. on p. 16).

[66] M. GB, M. WK, and M. RG. "A noise stress test for arrhythmia detectors". In: *Computers in cardiology* 11 (1984), pp. 381–384. URL: https://physionet.org/content/nstdb/1.0.0/ (cit. on pp. 16, 17).

[67] L. Huang et al. "Normalization Techniques in Training DNNs: Methodology, Analysis and Application". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (8 2023-08), pp. 10173–10196. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2023.3250241 (cit. on p. 18).

[68] S. Kaplan Berkaya et al. "A survey on ECG analysis". In: *Biomedical Signal Processing and Control* 43 (2018), pp. 216–235. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2018.03.003. URL: https://www.sciencedirect.com/science/article/pii/S1746809418300636 (cit. on p. 18).

I

Comparative Illustration of Traditional Methods Versus Proposed Model

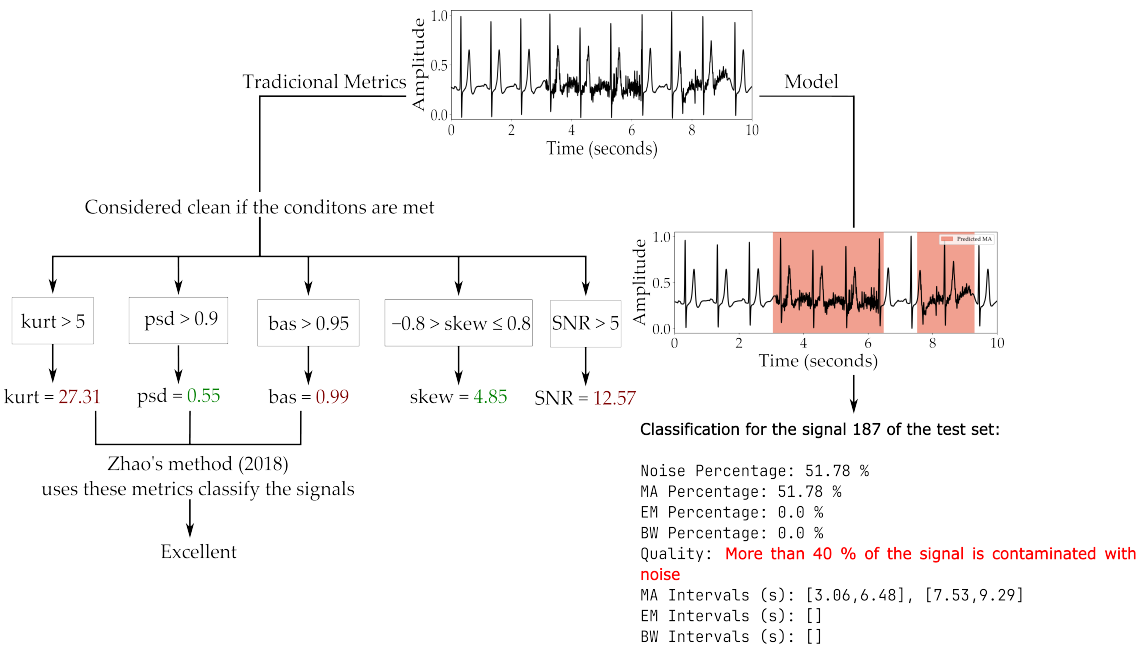


Figure I.1: Example of traditional metrics applied to signal 187 of the test set: The following SQI were applied to the presented signal: kurt, psd, bas, skew, and SNR. The thresholds for these indicators were derived from established methods as outlined in [10, 11]. A good quality signal usually is associated with kurt > 5, psd > 0.9, bas > 0.95, skew values between -0.8 and 0.8 and a SNR superior to 10 dB. A rule-based method by Zhao et al. was used to assess the overall signal quality [11]. While Zhao's method ranked the signal as "Excellent," this contradicts the visual appearance of the signal, as two of the five metrics, psd and skew, indicate a noisy signal. On the left, the signal processed by the employed model is shown for comparison.