



Master of Science Thesis

Lightweight Model Development for Arrhythmia Detection in Wearable Devices Using Deep Learning

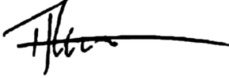
Student: Karageorgiou Stefanos
Registration Number: AIDL-0019

MSc Thesis Supervisor

Kasnesis, Panagiotis
Dr.

ATHENS-EGALEO, September 2024

This MSc Thesis has been accepted, evaluated and graded by the following committee:

| Supervisor | Member | Member |
|---|---|---|
|  | | |
| Kasnesis, Panagiotis | Patrikakis, Charalampos | Leligou, Eleni |
| Lecturer | Professor | Professor |
| Faculty Of Electrical and Electronics Engineering | Faculty Of Electrical and Electronics Engineering | Faculty Of Industrial Design and Production Engineering |
| University of West Attica | University of West Attica | University of West Attica |

Copyright © All rights reserved.

University of West Attica and Stefanos Karageorgiou
September, 2024

You may not copy, reproduce or distribute this work (or any part of it) for commercial purposes. Copying/reprinting, storage and distribution for any non-profit educational or research purposes are allowed under the conditions of referring to the original source and of reproducing the present copyright note. Any inquiries relevant to the use of this thesis for profit/commercial purposes must be addressed to the author.

The opinions and the conclusions included in this document express solely the author and do not express the opinion of the MSc thesis supervisor or the examination committee or the formal position of the Department(s) or the University of West Attica.

Declaration of the author of this MSc thesis

I, Stefanos Karageorgiou of Christos with the following student registration number: AIDL-0019, postgraduate student of the MSc program in “Artificial Intelligence and Deep Learning”, which is organized by the Department of Electrical and Electronic Engineering and the Department of Industrial Design and Production Engineering of the Faculty of Engineering of the University of West Attica, hereby declare that:

I am the author of this MSc thesis and any help I may have received is clearly mentioned in the thesis. Additionally, all the sources I have used (e.g., to extract data, ideas, words or phrases) are cited with full reference to the corresponding authors, the publishing house or the journal; this also applies to the Internet sources that I have used. I also confirm that I have personally written this thesis and the intellectual property rights belong to myself and to the University of West Attica. This work has not been submitted for any other degree or professional qualification except as specified in it.

Any violations of my academic responsibilities, as stated above, constitutes substantial reason for the cancellation of the conferred MSc degree.

The author

Stefanos Karageorgiou



Acknowledgements

I would like to thank Dr Panagiotis Kasnesis and Lazaros Toumanidis for their support and valuable advice throughout the dissertation. Finally, I dedicate this work to my parents who support me unconditionally in every step that I make.

Abstract

The increasing prevalence of wearable health devices has opened new possibilities for continuous, real-time monitoring of critical medical conditions, such as arrhythmias. This thesis investigates the development of lightweight deep learning models for arrhythmia detection, optimized for deployment on wearable devices. The main objective was to create models capable of detecting a variety of cardiac abnormalities from 12-lead ECG signals while maintaining a balance between diagnostic accuracy and computational efficiency. The research utilized the PhysioNet/Computing in Cardiology Challenge 2020 dataset, consisting of over 40,000 ECG recordings, and focused on 27 key arrhythmias. The methodology involved developing and evaluating several deep learning architectures, including convolutional neural networks (ECGConvNet) and hybrid models combining convolutional layers with long short-term memory (ECGConvLSTMNet). These models were trained on ECG data and evaluated using a custom evaluation metric to assess classification accuracy. To optimize the models for wearable devices, post-training quantization was applied, reducing model size while preserving as much accuracy as possible. The performance of the quantized models was compared with non-quantized models to understand the trade-offs between size, speed, and accuracy.

The results indicate that while the quantized models maintain a good level of accuracy, there is still room for improvement, particularly in handling more complex arrhythmias. The final quantized model achieved an F1-macro score of 0.264 on the validation set, showing the feasibility of deploying lightweight deep learning models for real-time ECG monitoring on wearable devices. However, limitations in computational resources and the complexity of ECG data imposed certain constraints on the research, including limited training time due to CPU-based processing and challenges in addressing underrepresented arrhythmias in the dataset.

In conclusion, this thesis demonstrates that lightweight deep learning models for arrhythmia detection in wearable devices are promising but require further refinement. Future research should explore advanced quantization techniques, such as quantization-aware training, and investigate additional architectures like transformers to enhance performance. The findings of this research contribute to the growing field of wearable health technology and have the potential to improve early detection and management of cardiovascular diseases through continuous real-time monitoring.

Keywords

Arrhythmia detection, deep learning, ECG classification, wearable devices, model quantization, real-time prediction.

Table of Contents

| | |
|--|-----------|
| Acknowledgements | 4 |
| List of Tables | 8 |
| List of figures | 8 |
| Acronym Index | 9 |
| 1 Introduction | 11 |
| 1.1 The subject of this thesis | 11 |
| 1.2 Aim and objectives | 11 |
| 1.3 Structure..... | 11 |
| 2 Electrocardiograms | 13 |
| 2.1 Cardiovascular Diseases and Their Global Burden..... | 13 |
| 2.2 The Role of Early Diagnosis in the Prevention of Cardiovascular Deaths | 13 |
| 1.2.3 ECG: Terminology and Function | 14 |
| 2.3 The Usefulness of Early Prevention with ECGs..... | 14 |
| 2.4 The Electrocardiogram: Traditional Procedure and Function | 14 |
| 2.5 The 12-Lead ECG: A Close-Up Look at Heart Activity | 16 |
| 3 Machine Learning and Deep Learning in ECG Classification | 18 |
| 3.1 Conceptual and Technical Overview of Machine Learning and Deep Learning | 18 |
| 3.2 Deep Learning in Medical Diagnostics and ECG Interpretation | 19 |
| 3.3 Deep Learning Architectures for ECG Classification | 20 |
| 1.3.3 Convolutional Neural Networks (CNNs)..... | 20 |
| 2.3.3 Long Short-Term Memory Networks (LSTMs)..... | 21 |
| 3.3.3 Wide and Deep Networks | 22 |
| 3.4 Multi-Label Classification in ECG Interpretation..... | 23 |
| 4 Real-Time Prediction and Lightweight Models for Wearable Devices | 25 |
| 4.1 The Importance of Real-Time Prediction in Medical Diagnostics | 25 |
| 4.2 Wearable Devices and Lightweight Models..... | 25 |
| 4.3 Model Quantization: Techniques and Impact on Accuracy..... | 26 |
| 5 Data | 28 |
| 5.1 Overview of the PhysioNet Dataset..... | 28 |
| 6 Literature Review | 35 |
| 7 Data Preprocessing | 39 |
| 7.1 Overview of the Preprocessing Pipeline | 39 |
| 7.2 Data Loading and Resampling..... | 39 |
| 7.3 Filtering | 40 |
| 7.4 Signal Normalization..... | 40 |
| 7.5 Random Window Extraction..... | 40 |
| 7.6 Demographic Feature Processing | 41 |
| 7.7 Target Variable Handling | 41 |
| 8 Methods | 43 |
| 8.1 Baseline Convolutional Neural Network (CNN) Architecture..... | 43 |

| | | |
|------------|--|-----------|
| 8.2 | Convolutional and Long Short-Term Memory (ECGConvLSTMNet) | |
| | Architecture | 44 |
| 8.3 | Wide and Deep Neural Network (WideAndDeepECGNet) | 45 |
| 8.4 | Enhanced Convolutional Neural Network (ECGConvNet) | 45 |
| 8.5 | Quantized model (QECGConvNet) | 46 |
| 8.6 | Individual probability threshold tuning | 47 |
| 8.7 | Evaluation Results | 48 |
| 9 | Conclusions | 51 |
| 9.1 | Key Findings | 51 |
| 9.2 | Limitations | 52 |
| 9.3 | Future Work | 52 |
| | Bibliography – References – Online sources | 53 |
| | Appendix: MAX78000 and ai8x Module for Model Quantization | 58 |

List of Tables

Table 5.1: Distribution of records between sources

Table 5.2: Population demographics across data sources

Table 8.7: Evaluation results compared to the literature

List of figures

Figure 2.1 Normal inner walls of coronary artery (left), versus artery with atherosclerosis (right). (Medical gallery of Blausen Medical 2014; WikiJournal of Medicine 1 ISSN 2002-4436. CC BY 3.0)

Figure 2.4.1: The illustration of a typical ECG exam (From: <https://www.mountsinai.org/health-library/tests/electrocardiogram>)

Figure 2.4.2: ECG waveform (From Understanding the EKG Signal: <https://a-fib.com/treatments-for-atrial-fibrillation/diagnostic-tests-2/the-ekg-signal/>)

Figure 2.5: Illustration of all the angles captured by a 12-lead ECG (From: <https://www.cablesandsensors.eu/pages/12-lead-ecg-placement-guide-with-illustrations>)

Figure 3.3.1: Illustration of a typical convolutional operation inside a CNN (From: https://www.researchgate.net/figure/Schematic-diagram-of-a-basic-convolutional-neural-network-CNN-architecture-26_fig1_336805909)

Figure 3.3.2: Illustration of an LSTM block (From: <https://databasecamp.de/en/ml/lstms>)

Figure 5.1: Pie chart of records' distribution across various sources

Figure 5.2: Gender distribution

Figure 5.3: Age distribution per gender group and outlier detection

Figure 5.4: A countplot of all the 111 classes available in the dataset

Figure 5.5: A countplot of all the 27 selected classes available in the dataset

Figure 5.6: A countplot of the number of labels per ECG record

Figure 7.1: Visual representation of raw 12-lead ECG record from our training set]

Figure 7.2: Visual representation of raw 12-lead ECG record from our training set after preprocessing

Figure 8.7: Reward matrix proposed by the challenge [50]

Acronym Index

1st degree AV block (IAVB): First-degree Atrioventricular Block

AF: Atrial Fibrillation

AFL: Atrial Flutter

Brady: Bradycardia

CRBBB: Complete Right Bundle Branch Block

IRBBB: Incomplete Right Bundle Branch Block

LAnFB: Left Anterior Fascicular Block

LAD: Left Axis Deviation

LBBB: Left Bundle Branch Block

LQRSV: Low QRS Voltages

NSIVCB: Nonspecific Intraventricular Conduction Disorder

PR: Pacing Rhythm

PAC: Premature Atrial Contraction

PVC: Premature Ventricular Contractions

LPR: Prolonged PR Interval

LQT: Prolonged QT Interval

QAb: Q Wave Abnormal

RAD: Right Axis Deviation

RBBB: Right Bundle Branch Block

SA: Sinus Arrhythmia

SB: Sinus Bradycardia

NSR: Sinus Rhythm

STach: Sinus Tachycardia

SVPB: Supraventricular Premature Beats

TAb: T Wave Abnormal

TInv: T Wave Inversion

VPB: Ventricular Premature Beats

CAD: Coronary Artery Disease

CNN: Convolutional Neural Network

CVD: Cardiovascular Disease

DL: Deep Learning

ECG: Electrocardiogram

FIR: Finite Impulse Response

LSTM: Long Short-Term Memory

ML: Machine Learning

RNN: Recurrent Neural Network

SE-ResNet: Squeeze-and-Excitation Residual Networks

QRS: QRS Complex (part of ECG signal)

P wave: Atrial depolarization (ECG component)

T wave: Ventricular repolarization (ECG component)

WHO: World Health Organization

1 Introduction

In recent years, the growth of wearable technology has transformed how healthcare is delivered, particularly in real-time health monitoring. Wearable devices, such as smartwatches, have become increasingly capable of detecting a range of medical conditions, offering continuous monitoring and early detection of life-threatening ailments. One such condition, arrhythmia—an irregularity in the heart’s rhythm—can lead to serious health complications if not detected early. This thesis investigates the feasibility of developing lightweight deep learning models for arrhythmia detection, designed for implementation in wearable devices to provide real-time health monitoring.

1.1 The subject of this thesis

The central issue addressed in this thesis is whether a deep learning model can be developed and optimized for arrhythmia detection and subsequently compressed to a lightweight version suitable for wearable devices. Arrhythmia detection presents significant challenges due to the complexity of electrocardiogram (ECG) signals and the need for accuracy in real-time scenarios. The motivation for this research is both timely and important, as wearable health technology is gaining traction, offering non-invasive, continuous health monitoring that could greatly impact early diagnosis and management of cardiovascular diseases.

1.2 Aim and objectives

The aim of this thesis is to develop a deep learning model for arrhythmia detection, quantize it for deployment on wearable devices, and evaluate its performance in terms of accuracy and resource efficiency. The specific objectives are:

- To explore different deep learning architectures suitable for ECG classification.
- To train and evaluate models on a large, multi-label ECG dataset.
- To apply quantization techniques to reduce model size without significantly compromising accuracy.
- To assess the feasibility of deploying these lightweight models in real-time wearable applications.

1.3 Structure

The structure of this thesis is organized as follows: Chapter 2 introduces electrocardiograms (ECGs) and discusses the global burden of cardiovascular diseases. Chapter 3 reviews the role of machine learning and deep learning in ECG classification, focusing on key architectures. Chapter 4 explores the importance of real-time prediction and the challenges associated with developing lightweight models for wearable technology. In Chapter 5, the dataset used in the thesis as well as in the literature is presented and an explanatory data analysis is conducted. Chapter 6 provides a detailed literature review on arrhythmia detection using deep learning models. Chapter 7 outlines the methods employed for data preprocessing in our thesis. Chapter 8 describes the development of the deep learning models, including the quantization techniques

applied. Finally, Chapter 9 concludes the thesis by summarizing the findings and outlining potential directions for future work.

2 Electrocardiograms

In this section, we introduce the underlying basics of electrocardiograms (ECGs) and their interpretation. We start by describing the severity of cardiovascular diseases and highlight the importance of early diagnosis. We then deep dive into how the process of capturing an electrocardiogram works and how a clinician can interpret the results. Moreover, we elaborate on the differences between regular and 12-lead ECGs along with their clinical significance.

2.1 Cardiovascular Diseases and Their Global Burden

According to the World Health Organization [1], Cardiovascular Diseases (CVDs) have now become the world's leading cause of mortality. An estimation of 17.9 million deaths from CVDs per year is projected, mainly from heart attacks and strokes, accounting for 31% of all global deaths. Among these causes, one of the most common is coronary artery disease (CAD). CAD involves the narrowing or blockage of the coronary arteries supplying oxygen-rich blood to the heart muscle due to atherosclerosis, which is a buildup of plaque inside the artery walls as shown in Fig [2.1]. As a result, this may cause chest pain, often described as angina, a heart attack, or even sudden death [2].

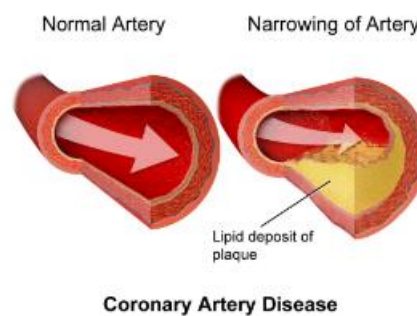


Figure 2.1: Normal inner walls of coronary artery (left), versus artery with atherosclerosis (right) (Medical gallery of Blausen Medical 2014; WikiJournal of Medicine 1 ISSN 2002-4436. CC BY 3.0)

It is estimated that tens of thousands of people die from some form of heart condition daily around the world [3]. In developed countries, coronary artery disease is the leading cause of death, while in the low- and middle-income countries, the rapidly increasing rates of death from heart conditions are often related to changes in lifestyle, involving unhealthy diets, lack of physical activity, and smoking [4]. The CVDs result in a massive burden on mortality and economic costs for treatment and loss of productivity [5].

2.2 The Role of Early Diagnosis in the Prevention of Cardiovascular Deaths

Early diagnosis of heart abnormalities is one of the easiest ways to avoid severe cardiac events like myocardial infarction and heart failure. One of the most efficient, non-invasive diagnostic tools for this purpose is an electrocardiogram (ECG). An ECG is one of the most common diagnostic tools that records any electrical activity related to the heart over a period of time [6]. It helps in identifying various types of arrhythmias, ischemic heart disease, and other disorders that may indicate hidden cardiovascular diseases [7].

The sooner these abnormalities are picked up, the better the chances are for effective interventions. Early prevention by frequent ECG screening is very useful, especially in high-risk cases, including subjects with a medical history of CVD, hypertension, diabetes, and/or high cholesterol [8]. Most of the benefits from ECGs in early prevention are derived from the fact that doctors can identify electrical signals in the heart that indicate a predisposition to problems through such procedures during the early stages of a disease, before worse symptoms appear [9].

1.2.3 ECG: Terminology and Function

ECG stems from ElectroCardioGram; this is a word derived from "electro," as in electrical activity, and from "cardio," as in heart related. The electrocardiogram records electrical impulses produced by the movement of the heart and displays these impulses on a graph in waveform. These waveforms allow clinicians to analyze the rhythm of the heartbeat, assess electrical conduction problems, and diagnose a variety of cardiovascular conditions.

2.3 The Usefulness of Early Prevention with ECGs

The early prevention in the management of cardiovascular diseases is of high importance. The identification of a modality of cardiovascular origin has been proved in many studies to be much better when it was recognized early [10]. For example, early diagnosis of atrial fibrillation, ventricular hypertrophy, and ischemic heart disease may significantly reduce the risk of complications related to stroke, heart failure, or sudden cardiac death [11].

ECGs have become central in such a preventative strategy. For example, patients with high risks for cardiovascular diseases, such as those with hypertension, diabetes, or even family history of heart disease, can be regularly screened through ECGs [12]. These examinations will help the clinician monitor the first signs of cardiac dysfunction even when symptoms are not apparent. For example, the onset of ischemia or hypertrophy can often be represented by very slight changes in the P wave, QRS complex, or ST segment (ECG components that are mentioned in the next section), and may lead to more serious conditions [9]. Health professionals can thus begin with the relevant treatments, from anticoagulants for atrial fibrillation to beta-blockers for hypertension, in order not to let the disease advance any further.

Moreover, the ECG is also useful in follow-up on the treatment outcomes. For instance, ECGs could be repeatedly carried out for patients who have undergone procedures such as angioplasty or bypass surgery to ensure that no new complication has arisen [13]. In this way, beyond early detection, the ECG could also play a vital role in the long-term management and prevention of cardiovascular diseases [6].

2.4 The Electrocardiogram: Traditional Procedure and Function

The electrocardiogram records the electrical activity of the heart through electrodes attached to the patient's skin as illustrated in Fig [2.4.1]. These electrodes collect the electrical signals generated by the beating heart and transmit these to the ECG machine, which in turn prints out a graphical recording of the heart's activity [14]. The basis of a typical ECG tracing is the P wave, representing the depolarization of the atria; the QRS complex, representing ventricular

depolarization; and a T wave, relating to the repolarization of the ventricles. These waveform studies might achieve differential diagnoses related to various heart disorders, including arrhythmias and myocardial ischemia [15].

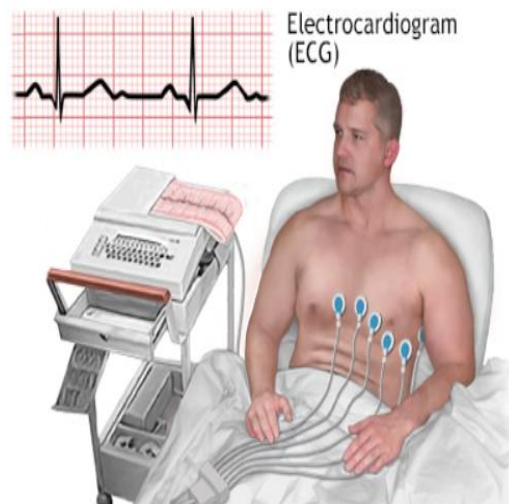


Figure 2.4.1: The illustration of a typical ECG exam (From: <https://www.mountsinai.org/health-library/tests/electrocardiogram>)

In a routine ECG, electrodes are placed on the chest, arms, and legs. These electrodes are connected to the ECG recorder, which charts electrical impulses as they move through the heart [16]. Consequently, electrical activity will be reflected as a series of waves and intervals on the ECG tracing. Any abnormality in the pattern of these waves and intervals can indicate some heart condition. For instance, ST segment (Fig [2.4.2]) above normal would indicate an acute myocardial infarction; T wave with abnormal characteristics may also signal ischemia or even abnormal electrolyte levels [17].

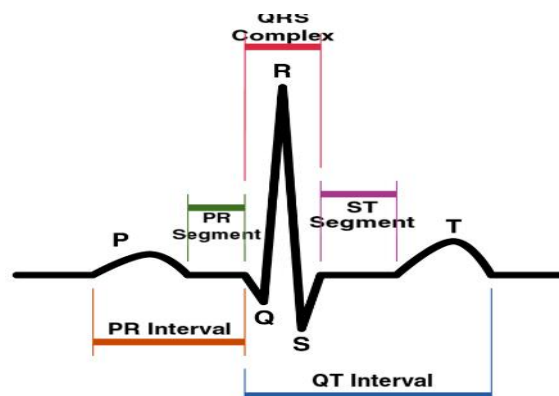


Figure 2.4.2: ECG waveform (From Understanding the EKG Signal: <https://a-fib.com/treatments-for-atrial-fibrillation/diagnostic-tests-2/the-ekg-signal/>)

While the process of a routine ECG is rather elementary, the placing of the electrodes must be very precise; otherwise, it will give a misleading reading. Poor placing gives an incorrect impression of the electrical activity of the heart that may lead to failure or misdiagnosis of a

problem. Therefore, proper training and experience are crucial not only for the technician carrying out the ECG but also for the clinician when interpreting the results [18].

2.5 The 12-Lead ECG: A Close-Up Look at Heart Activity

The 12-lead electrocardiogram is the most developed kind of ECG and finds wide application in clinical practice, targeting a detailed observation of electrical activity of the heart. While simpler ECG configurations record only from a single or few leads, this 12-lead ECG captures the electrical signals of the heart from 12 different angles and gives a much more complete picture of cardiac function (Fig [2.5]) [16].

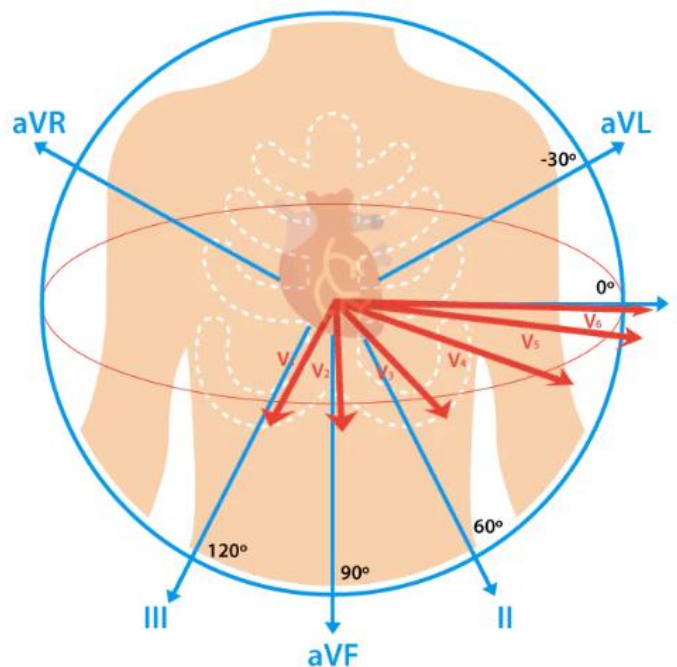


Figure 2.5: Illustration of all the angles captured by a 12-lead ECG (From: <https://www.cablesandsensors.eu/pages/12-lead-ecg-placement-guide-with-illustrations>)

ECG leads have traditionally been divided into two types: limb leads and precordial leads. Limb leads refer to the electrical axis of the heart in the frontal plane (I, II, III, aVR, aVL, aVF), while precordial refers to its activity in the horizontal plane: precordial or V leads (V1 to V6) [12]. Each lead offers a different perspective—the electric waves emanating from different parts of the heart. For example, leads II, III, and aVF reflect the electrical activity of the inferior wall of the heart, whereas the leads V1 to V4 focus on the anterior wall [19]. This wide field-of-vision helps the doctor get abnormalities that may not be obvious in single lead ECG—for instance, localized myocardial infarctions or bundle branch blocks [15].

The 12-lead ECG is especially useful in the diagnosis of acute coronary syndromes, arrhythmias, and ventricular hypertrophy. The electrical activities received through the different leads allow

the clinician to determine the position and extent of any myocardial infarct, the status of any conduction block, or the status of the electrical conducting system of the heart in general [20]. These include the elevation of the ST segment in leads indicative of acute myocardial infarction and the implicated lead indicative of the area of the heart involved [21].

Besides the already mentioned, the 12-lead ECG is very important in emergency management for patients presenting either with chest pain or other symptoms of myocardial infarction. In such instances, the 12-lead ECG provides instantaneous, real-time information that guides lifesaving interventions, including thrombolytic therapy or urgent angioplasty [22]. The rapid diagnosis and localization of a myocardial infarction on a 12-lead ECG make it indispensable in both emergency and routine clinical practices [13].

To conclude, cardiovascular diseases still pose the greatest challenge to health in this century, claiming millions of lives every year. In relation to early diagnosis and management, an ECG-a more appropriate term is a 12-lead ECG-is primarily an early basic step. ECGs allow clinicians, through the provision of a close-up view of the electrical activity of the heart, to outline abnormalities early, sometimes even before symptoms have appeared, and institute necessary interventions with a view to preventing worse outcomes. Thus, the 12-lead ECG is the highest resolution record of the electrical function of the heart, as it takes electrical signals from many angles that provide very important information concerning rhythm and structure. Considering the burden of cardiovascular disease continuing to rise, ECGs will be at the forefront in diagnosis and prevention as far as minimizing mortality and improving outcomes for patients, a fact that makes the idea of their examination automation via artificial intelligence more attractive than ever.

3 Machine Learning and Deep Learning in ECG Classification

Deep learning (DL) has undeniably changed how difficult problems were envisioned lately, especially those domains that incorporate voluminous data processing and identification of complex patterns in them. Cardiology is a field where doctors must study and observe an ECG for diagnosing most heart diseases. Recent publication trends have shown that Machine Learning (ML) can be a powerful tool for various areas, including medicine. A variety of breakthroughs in medicine have been achieved by incorporating medical knowledge into algorithms, resulting in the development of tools able to help expert clinicians to make more advanced decisions and also automate and alleviate some of their mundane tasks. In this chapter a summary overview of both applied ML and DL frameworks in ECG classification is provided, thereby underlining technological groundings that could form the basis for the adoption of models in this thesis. We will start with an introduction to basic machine learning and deep learning concepts, followed by a discussion of specific architectures that have proven to be useful for dealing with the spatial and temporal complexities of ECG data, such as Convolutional Neural Networks, Long Short-Term Memory Networks, and Wide and Deep Networks. Finally, the methodological approach to multi-label classification will be briefly covered, which puts a basis, essentially from the same ECG data, on which several cardiac anomalies could be identified. Understanding the theoretical background of these methods would provide a perspective on how these methods are able to deliver accurate real-time predictions within wearable health devices and lay the bedrock for the next methods and experiments of the thesis.

3.1 Conceptual and Technical Overview of Machine Learning and Deep Learning

The domain of Artificial Intelligence (AI) which can be defined as the capability to enable machines to learn from data, identify patterns, and make judgments or predictions autonomously is called Machine Learning. It is based on the representation and execution of algorithms which analyze the input data to find patterns within, then use those inferences on other unobserved data [23]. Supervised learning is considered one of the most common approaches toward machine learning where the model has to be trained with labeled data. In other words, that would mean that both input and output labels are available; thus, an algorithm learns the mapping from input to output [24]. In our thesis scope, using the pattern in electrocardiogram waveforms, machine learning models can be taught to recognize certain prespecified heart problems [25].

Machine learning can be demarcated from other traditional modes of programming in that it relies on data for building features and decision boundaries. Conventionally, rule-based systems are hand-crafted by a human expert by laboriously creating rules for decision-making. While instead, in ML models, these rules are automatically learned from data by the maximization of objective functions, and the performance in these models depends upon the quality and size of the training data [26].

Deep learning is a subset of several different machine learning methods, where a lot of academic articles in the literature would parallelize that the inspiration of their architecture comes from the structure and functionality of the human brain. That is, deep learning models are made to intentionally learn hierarchical input representations automatically from complex links and patterns deriving from the raw input of the data. These models are referred to as neural networks and contain various interconnected nodes or neurons in successive layers, which transform the inputs in an increasingly abstract manner [27]. Probably the most important ability of deep learning that makes it so powerful in picture and signal processing, including electrocardiogram processing, is its ability to learn these representations mentioned above, from raw data[28].

The process in which, in deep learning, a model learns its parameters, especially the weights of its interneuron connections, is known as backpropagation. In an overly broad explanation, it's the act of having the weights iteratively modified based on the gradient of a loss function. In simple terms, the event chain goes as follows: prediction on the training data, calculation of error/loss, and backpropagation of that mistake through the network to adjust the weights. The entire process is repeated many times, for in several epochs until convergence occurs and a minimum error is obtained [29].

Deep learning models are suitable for handling large-sized datasets, and can thus be exploited to complex tasks like the ECG classification presented with multidimensional, noisy, and temporally varying signals. This is especially true for tasks such as arrhythmia classification [25], where minute changes in the waveform could spell the thin line between life and death. Deep learning architectures allow one to build systems whereby such patterns from the data are themselves learned by the model with superior performance compared to traditional rule-based and feature-engineered techniques.

3.2 Deep Learning in Medical Diagnostics and ECG Interpretation

Deep learning now ingrains analysis for complex physiological data in cardiology. As mentioned in chapter 2, an elementary diagnostic tool in cardiology, the 12-lead ECG, records the electrical activity of the heart from multiple perspectives and thus gives the doctor an all-encompassing look into cardiac function. Each lead of the ECG traces electrical potentials between two pre-specified pairs of electrodes, giving another different perspective on the electrophysiology of the heart. By incorporating information from all 12 leads, physicians are able to diagnose abnormalities such as cardiac arrhythmias and conduction blocks.

Traditionally, ECGs have been interpreted based on the cardiologist's expertise in observing waveforms for anomalies. This approach is time-consuming and prone to human error, especially when minute abnormalities or inconsistencies across leads are present. Deep learning ECG classification seeks to automate this process, using raw waveforms to arrive at highly accurate diagnoses [25].

The challenge in ECG analysis stems from its inherent complexity, as the waveform represents numerous physiological processes. and an abnormality may show up only as minor variations

in some segments of the signal. For instance, an abnormal P wave may suggest atrial fibrillation, while an irregular QRS complex could indicate bundle branch blocks [28]. This is where deep learning excels: its ability to model complex relationships and learn hierarchical representations makes it highly effective for analyzing these nuanced variations in ECG data [27].

3.3 Deep Learning Architectures for ECG Classification

The success of deep learning in ECG classification depends on the choice of architecture, each of which is optimized for handling different aspects of the data. There is a plethora of major architectures that are being put into practice in ECG classification over the past years which include CNNs, LSTMs, and more recently very popular ones like Wide and Deep Networks [30]. Each architecture reflects different strengths in interpreting ECG signals, which exhibit both spatial and temporal complexities. CNNs excel at learning spatial features from raw ECG signals, capturing local patterns and waveform shapes, while LSTMs are well-suited for learning temporal dependencies, such as detecting arrhythmias across time sequences [25]. Wide and Deep Networks combine the benefits of wide linear models and deep neural networks, making them effective in learning both low- and high-dimensional patterns [31]. In this chapter an overview of each architecture will be provided emphasizing on their potential usage in the context of ECGs.

1.3.3 Convolutional Neural Networks (CNNs)

CNNs were designed for data with grid-like structures and are therefore eminently suitable for picture or time-series data, a feature that makes them very effective in ECG signal processing. Generally speaking, a convolutional neural network works by applying convolutional filters on the input data. In such a way, it will be able to detect local patterns within the signal, like edges, shapes, or any other typical waveform [26]. Through the development of the work with ECG, these filters can be empowered to select clinically relevant features such as the P wave, QRS complex, and T wave, important for the diagnosis in cases of cardiac disorders [30].

In a 12-lead ECG, each lead represents one channel in this time series signal. CNNs have the capability of processing more than one channel at once and can extract the relationships in the spatial features spanning multiple leads. This will be the case in which some arrhythmias look different across leads; the key to identification lies in exact diagnosis [28]. These CNNs systematically extract advanced information from the ECG by combining many convolutional layers, including low-level fluctuations of the signal with higher-order interactions between different leads.

The typical architecture of a CNN involves pooling by max-pooling or average-pooling (Fig [3.3.1]); this reduces the feature maps in dimensionality while keeping the most important information. While doing so, it decreases the problem's computational complexity and enhances the robustness of the model against small variations of the input signal [32]. Lastly, after these convolution and pooling steps, the feature maps usually undergo flattening and fully connected layers connecting the extracted knowledge to produce the estimated target in classification.

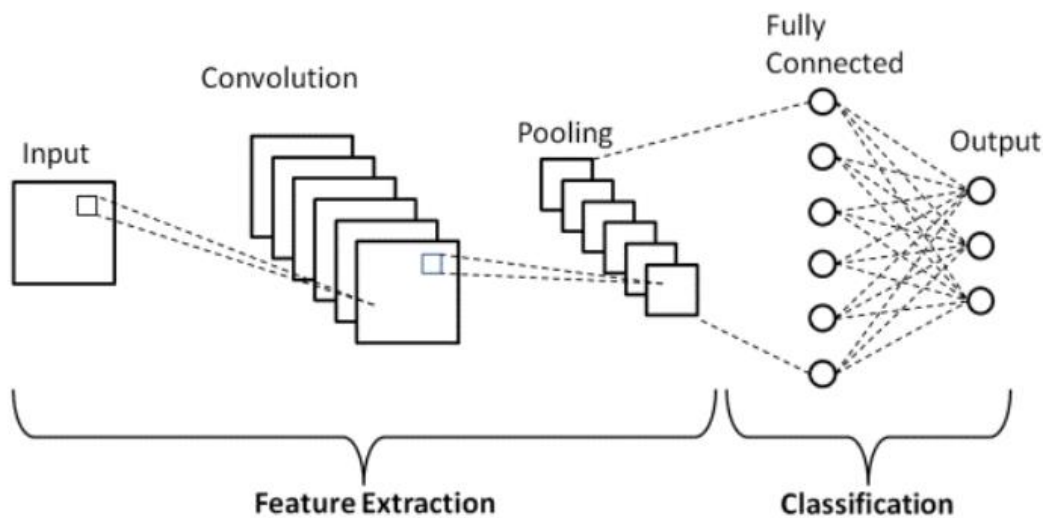


Figure 3.3.1: Illustration of a typical convolutional operation inside a CNN (From: https://www.researchgate.net/figure/Schematic-diagram-of-a-basic-convolutional-neural-network-CNN-architecture-26_fig1_336805909)

Other benefits of CNNs include great efficiency, especially in real-time applications. The speed with which it does the inference after it has been trained has proven to be faster compared to other architectures that rely on fully connected neurons and layers, a characteristic that is quite desirable for the real-time analysis of ECGs on wearables. Finally, their ability to process volumes of data also places them clinically for the management of a number of ECGs all at once [33].

2.3.3 Long Short-Term Memory Networks (LSTMs)

While CNNs proved capable of catching those spatial patterns in the ECG signal, they tend to capture little to none temporal dependencies because of their architecture's nature. It is here that the LSTMs, a subclass of Recurrent Neural Networks (RNNs), come into relevance. Their architecture allows for a more accurate way of processing sequences of data by remembering previous time steps-as the "memory" helps in the detection of longer sequence dependencies [34].

In fact, the transient nature of ECG signals makes LSTMs be of special use in applications such as arrhythmia identification, where the diagnostic information usually resides not in the single pulse but in a sequence of beats. As an example, AF and STach are rhythm disorders; they manifest through sustained cardiac rhythm irregularities, a fact that highlights even further the importance of an architecture capable of capturing the time essence into the data [28]. As previously mentioned, LSTMs cache and learn the temporal properties of the data by preserving information from earlier in the sequence context to inform the predictions later.

Every LSTM cell contains three major parts: the input gate, forget gate, and output gate as shown in Figure [3.3.2]. These gates allow for the flow of information in and out of the cell, thereby

enabling the network to discard unnecessary information and store only the necessary information. The aforementioned statement is what makes LSTMs powerful, since their architecture helps them avoid the issue of vanishing gradients, a major drawback of traditional RNNs, and hence learn the long-term dependencies quite easily [35].

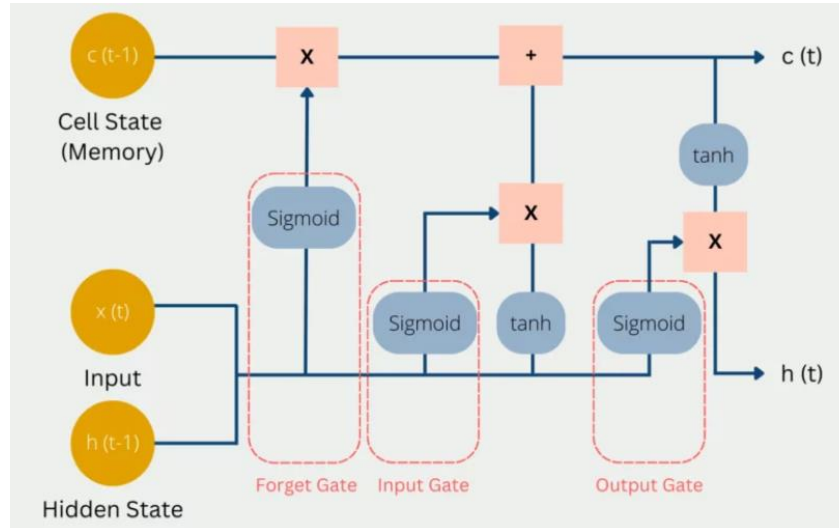


Figure 3.3.2: Illustration of an LSTM block (From: <https://databasecamp.de/en/ml/lstms>)

In real-world applications, LSTMs are often combined with CNNs in hybrid models that leverage the strengths of each architecture. Usually, the CNN will be used to extract the spatial information from the ECG data, while the LSTM will manage the temporal relationships, hence allowing the model to take in not just the spatial structure of the individual waveforms but the temporal sequence of heartbeats. This model fusion has proven especially effective in the domain of arrhythmia recognition, many forms of which need both spatial and temporal contexts to be diagnosed well [33].

3.3.3 Wide and Deep Networks

Wide and Deep Networks are a new paradigm for deep learning that incorporates the strengths of both shallow architectures comprising wide networks and deep architectures. The notion behind Wide and Deep Networks lies into the combination of both structured and unstructured data. The “wide” part of the model learns those patterns and interactions which are explicitly defined by the features, whereas the “deep” part learns patterns by leveraging the deep learning capabilities from raw unstructured data hierarchically. This architecture is particularly fitted for the task of ECG classification because the model can make use of both handcrafted features-including specific waveform morphologies-and features that are automatically learned, such as complex patterns of the time series [31].

The wide part resembles most of the more classical machine learning models, whereby the explicit features are supplied directly to the model as an input. Generally, such features can be hand-crafted or based on domain expertise and are meant to give the model a "shortcut" toward finding specific patterns already known to be diagnostically relevant [30].

It is the deep component, however, that processes the raw ECG signal with many convolutional or recurrent structure layers and brings up the patterns that a human annotator may be unable to produce. Capturing this representation hierarchically from the data produces more abstract and complex features than it could easily get with the mere adoption of the pre-defined features [33]. Therefore, Wide and Deep Networks can learn both pre-defined and new relationships of the ECG data, improving results in this respect.

Wide and Deep Networks are particularly valuable in scenarios where certain features are well understood, but there is also a need for discovering new, latent patterns within the data. In ECG classification, this could mean that while certain arrhythmias are well-defined by traditional metrics, other, more subtle abnormalities may require deeper exploration through learned representations. The combination of both wide and deep components ensures that the model benefits from domain knowledge while also allowing it to learn new patterns from the data [28].

3.4 Multi-Label Classification in ECG Interpretation

Now it is a good moment in the thesis to note that the classification of ECGs is inherently multi-label: one ECG tracing can be associated with several diagnostic categories. A patient can have both AF and LBBB; correspondingly, a model should also be able to identify the presence of both illnesses simultaneously. This problem is considerably harder compared to traditional single-label classification, where for one sample, only one label is assigned.

The ECG analysis follows a multi-label classification setting where the model has to predict a binary outcome for each of the possible diagnoses. That is, on a given ECG recording, the model will return a probability for each class, showing whether the sickness is present or not. Commonly, these probabilities are thresholded to give binary predictions, where a probability above some threshold-0.5, for example-indicates that the condition is present [28].

Training multi-label classification involves the optimization of a loss (or objecting) function that takes into consideration all the predictions for every label. In that respect, binary cross-entropy is usually used as the loss function. It calculates an error for every label separately and then sums them up. In that way, the model can handle multiple labels all at once and give very good predictions for both frequent and rare conditions [33].

Precision, recall, and F1 score, are some of the metrics that have been applied to quantify the performance of multi-label classifiers. The approach captured above provides a view of the model performance across multiple classes and allows the making of trade-offs between false positives and false negatives. This is because a few disorders may be highly prevalent and thus the model may easily be overfitted towards the majority classes at the cost of clinically important but very rare conditions [31].

Thus, deep learning offers a variety of architectures discussed above that can be utilized to capture spatial and temporal representations of the ECG data. Framing the problem as a multilabel classification problem can find multiple cardiac diseases from a single recording of ECG and develop more accurate, efficient, and scalable diagnostic systems. One topic remains

uncovered yet. While building highly accurate models for arrhythmia prediction is vital, the era of wearable devices that feature diagnostic models creates a new benchmark. This in turn opens a discussion on the significance of the inference and prediction time of such models on wearable devices. In the upcoming chapter we will discuss the importance of real time prediction on wearable devices and the methods to achieve it.

4 Real-Time Prediction and Lightweight Models for Wearable Devices

As wearable technology continues to evolve, the demand for real-time, low-latency medical diagnostics is becoming increasingly prominent. Wearable devices, such as smartwatches and fitness trackers, have transitioned from simple health monitoring tools to sophisticated diagnostic platforms capable of detecting critical medical conditions in real time. One of the key challenges in the design of such platforms relates to finding harmony between computing efficiency and prediction accuracy in activities such as arrhythmia detection with ECG data. This chapter reviews real-time prediction, with a focus on lightweight deep learning models that can be deployed on wearable devices. Finally, a set of techniques to reduce a machine learning model's size also known as quantization is introduced. Even though the literature is growing by the minute with novel quantization approaches, we will be reviewing some of the most important ones that help on reducing a model's size and complexity, hence making them suitable for real-time applications with minimal alteration in their diagnostic accuracy.

4.1 The Importance of Real-Time Prediction in Medical Diagnostics

Real-time prediction includes the ability of the system to process the incoming data and come out with output or diagnosis with little time being wasted. Some of the major motives behind the real-time prediction capability of wearable health devices are timely interventions that can be provided in life-threatening conditions. Whether it is a capacity to detect, within a few seconds, the onset of a dangerous arrhythmia-say, ventricular fibrillation-the wearable device could phone a lifesaving call to the user or health professional; it is really the speed of the response compared to conventional diagnostics that usually have huge delays between data acquisition, interpretation, and diagnosis [36].

Apart from the life threatening scenarios, real time prediction improves the user experience through making such predictions real-time, continuous, and actionable. Wearable devices are meant to be worn all the time, under some tracking of physiological signals such as heart rate, oxygen saturation, and electrocardiogram data. Wearable devices, operating under a batch analysis of data or after some considerable period, would make the results irrelevant since the findings would not attribute their existing time interval [37].

Real-time prediction technically requires an efficient and fast computational model. Although very accurate, deep learning models are usually very computation-intensive; hence, they require high-performance hardware to execute fast inference. Considering the limited processing power and memory, wearable devices can never achieve real-time performance; hence, there is an immediate need for lightweight dedicated models operating at a high level of diagnostic performance under strict hardware constraints within the wearables.

4.2 Wearable Devices and Lightweight Models

Wearable devices, including smartwatches, fitness trackers, and specialist medical wearables, have relatively few computational resources compared to traditional computing systems. Most of the devices are normally designed with a low-power processor, very constrained memory,

and a limited battery lifetime, which normally makes direct deployment of complex and large deep learning models challenging on the device. Hardware constraints call for lightweight model development that will realize accurate predictions without overburdening the device [39].

The term lightweight model can be described as a model with a small number of parameters, hence small in size that makes the time between the required calculations from the input step to the output step close to the second. Lightweight models were developed in an effort to decrease the computational and memory costs for deep learning models, so that it can easily enhance their on-the-fly use in resource-constrained environments. A few of the probable ways to design lightweight models include model compression, quantization, pruning, and distillation. Each of them shrinks a model's size through either removal of some of the superfluous parameters or through effective optimization of the model weight representations [39,40].

The techniques most applied today to reduce model complexity are network pruning, in which the model removes neurons and model connections that are contributing less or are redundant. This pruning can be performed during or after training, with the intention of discarding elements that do not add to the final prediction. Pruning reduces the total number of network parameters, hence reducing model memory usage [41].

Another applied strategy for this problem is model quantization, wherein both the weight and activation precision are reduced. Typically, most deep learning models are trained on 32-bit floating-point integers, but during resource-constrained deployment, this precision can be reduced to 16-bit, 8-bit, and even less without significant degradation in the model performance. This means that the model requires less memory and will do the computations much faster. It is thus suitable for real-time execution on wearables [42].

Model distillation in general is a process of training a smaller, simpler model-the "student"-to mimic the behavior of a larger, more complex model-the "teacher." The notion is to train a much smaller student model that will mimic the predictions of a much larger instructor model but will be times smaller and much lighter computation-wise. This method has been proven to be one of the most efficient approaches at creating very light models, while still holding excellent accuracy [43].

4.3 Model Quantization: Techniques and Impact on Accuracy

Quantization is one of the most common techniques that can be used to transform a complex, high-precision model into its lightweight version suitable for wearables. This process reduces the precision of the numerical values of the learned weights during the training of the model and hence allows faster computation and reduced consumption of memory [44]. Most deep learning models conventionally work on 32-bit floating-point base, which has very high precision but is quite computationally expensive. Quantization flows can reduce the computational load alone significantly by reducing precision to 16-bit or 8-bit integers with limited loss in accuracy.

There are numerous quantization algorithms, each making different trade-offs between model size, speed, and accuracy:

1. Post-Training Quantization: This is the application of quantizing once a model has been fully trained. In other words, a trained model is quantized by changing weights and activations from 32-bit floating-point representations into lower-precision formats. This is a somewhat simpler approach to use; however, this may result in some degree of loss in accuracy, especially for models that are very sensitive to even slight changes in the weights [45].

2. Quantization-Aware Training: Embedding the quantization process into the training phase can be one of the mitigations for avoiding post-training quantization accuracy loss. At QAT, the model emulates the quantization process in the forward passes of training, while allowing weight updates and full precision to adapt the lost accuracy. This will yield a far better model, maintaining its accuracy post-quantization. Although much more computationally intensive compared to direct post-training quantization, QAT usually results in a much better quantized model in practice [46].

3. Dynamic Quantization: This contains a technique wherein weights are quantized to lower precision while keeping the activations in high precision formats like that of a 32-bit floating point format. Hence, this balances the tradeoff between the decrease in memory and computational cost while mitigating losses in accuracy as far as possible. This is useful during instances when the model's accuracy depends upon sensitive changes in the precision of activation [47].

4. Integer-only quantization: This means that the weights and all the activations are transformed wholly into their integer representations. Correspondingly, this would give a model that will be able to use only integer arithmetic in its implementation, usually having pretty high speed compared with floating-point operations on low-power systems. However, in regard to integer-only quantization, large speedups and efficiency gains normally come at the price of higher degradation in model accuracy if not implemented carefully [48].

Eventhough quantization is indeed an effective technique that can be proposed for wearable devices, which are normally very limited in computational resources, there are a couple of challenges. The immediate trade-off is in achievable precision: Whereas most models can be quantized with very minor degradation in performance, some models do suffer, especially those with very complex features or when they are highly sensitive for even small changes in weights, hence turning can be very poor in accuracy after quantization [49]. These cases would have to be precariously calibrated and tested to ensure that they would still meet the required diagnostic limits.

5 Data

5.1 Overview of the PhysioNet Dataset

This thesis utilizes data from the PhysioNet/Computing in Cardiology Challenge 2020 [50], which provides a robust dataset of 12-lead electrocardiogram (ECG) recordings collected from a variety of data sources across different regions. The challenge aimed to stimulate advancements in the development of algorithms for automated cardiac abnormality detection, and it forms a critical foundation for developing deep learning models. The dataset's diversity in terms of geography, sampling methods, and patient demographics makes it particularly suitable for building generalizable models capable of arrhythmia detection in wearable devices. This chapter provides a detailed exploration of the dataset's composition, its demographic characteristics, and the focus on 27 cardiac abnormalities out of the total 111 diagnoses present in the dataset.

The training collection comprises ECG recordings from four primary sources, each reflecting distinct populations and healthcare environments. These data sources enhance the dataset's diversity regarding geography, clinical procedures, and patient demographics. The institutions contributing to this dataset are located in the United States, Europe, and Asia, guaranteeing that the data encompasses a wide range of cardiac diseases and patient demographics.

1. China Physiological Signal Challenge (CPSC): This dataset originates from the China Physiological Signal Challenge 2018 [51] and comprises ECG recordings obtained from a substantial cohort of patients in China. The dataset is partitioned into three sub-datasets: CPSC, CPSC-Extra, and a concealed test dataset designated for the Challenge's ultimate assessment.

2. St. Petersburg Institute of Cardiological Technology (INCART): This dataset, sourced from Russia, has a limited quantity of ECG recordings concentrated on arrhythmias. It offers high-resolution signals with extended recording times relative to other datasets [52].

3. Physikalisch-Technische Bundesanstalt (PTB and PTB-XL): These datasets, originating from Germany, comprise a basic version and an expanded version, PTB-XL, which offers more complete and extensive ECG recordings. The PTB-XL dataset is distinguished by its elevated sample rate and comprehensive metadata, encompassing demographic information [53].

4. The Georgia 12-lead ECG Challenge (G12EC) collection originates from Emory University in Atlanta, USA, and comprises recordings from a substantial population in the Southeastern United States. The G12EC dataset is divided into training, validation, and test sets, which substantially augment the overall size of the Challenge dataset.

The dataset includes 43,101 ECG recordings sourced from six different sub datasets and institutions across four countries. Each recording contains a 12-lead ECG signal saved in mat file format, with corresponding metadata such as the patient's age, gender, and diagnostic labels stored in hea files.

As shown in Table [5.1] and Figure [5.1], The PTB-XL dataset makes for over half of the total dataset. The Georgia and CPSC 2018 datasets contribute significantly as well, with large patient populations and high-quality ECG recordings. The smaller datasets, PTB and INCART, though limited in size, provide useful diversity in sampling rates and demographic profiles, which adds to the robustness of the dataset.

| Source | # of records | % of the total samples |
|-----------------------|--------------|------------------------|
| PTB-XL | 21,837 | 50.66% |
| Georgia | 10,344 | 24.00% |
| CPSC 2018 | 6,877 | 15.96% |
| CPSC 2018 Extra | 3,453 | 8.01% |
| PTB | 516 | 1.20% |
| St. Petersburg INCART | 74 | 0.17% |

Table 5.1: Distribution of records between sources

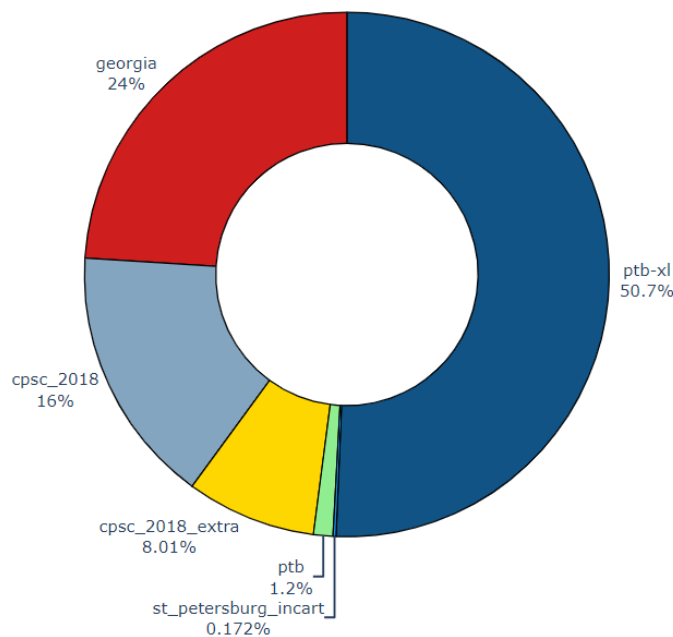


Figure 5.1: Pie chart of records' distribution across various sources

In addition to the ECG signals, the dataset contains demographic information about each patient, including gender and age. These features are important as they provide additional context for the ECG readings and can be incorporated into the deep learning model to improve classification accuracy. A summary of the demographic composition of the dataset is provided in the table below:

| Source | # Samples | % Females | % Males | Average Age | Sampling Rate |
|-----------------------|-----------|-----------|---------|-------------|---------------|
| CPSC 2018 | 6,877 | 46.21% | 53.79% | 60.13 | 500 Hz |
| CPSC 2018 Extra | 3,453 | 46.63% | 53.37% | 63.73 | 500 Hz |
| Georgia | 10,344 | 46.34% | 53.66% | 60.04 | 500 Hz |
| PTB | 516 | 26.74% | 73.06% | 55.31 | 1000 Hz |
| PTB-XL | 21,837 | 47.89% | 52.11% | 61.54 | 500 Hz |
| St. Petersburg INCART | 74 | 45.95% | 54.05% | 55.99 | 257 Hz |

Table 5.2: Population demographics across data sources

Across all hospitals, the gender distribution is relatively balanced, with 53% of the patients being male and 47% female. The total number of male patients is 22,889, while the total number of female patients is 20,211 as shown in Figure [5.2]. The average patient age ranges from 55.31 years in the PTB dataset to 63.73 years in the CPSC 2018 Extra dataset. The combination of both younger and older patients provides a dataset that mirrors the age-related prevalence of many cardiac conditions, such as atrial fibrillation, which is more common in older adults.

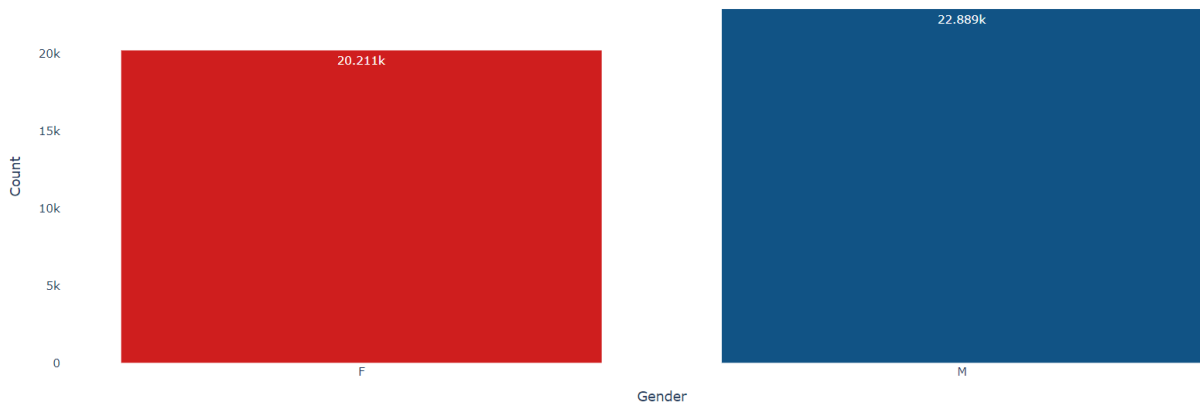


Figure 5.2: Gender distribution

Moreover, by looking at Figure [5.3] we can infer that the dataset has a wide age spectrum, predominantly featuring older persons, which indicates the heightened incidence of cardiac disorders like atrial fibrillation, bundle branch blockages, and ventricular arrhythmias among this demographic. If we also look closely we will notice outliers from both the upper and the lower tails of the distributions of both genders. That entails that data cleaning practices should take place in the preprocessing step of the data, before they reach out the training phase of our machine learning models.

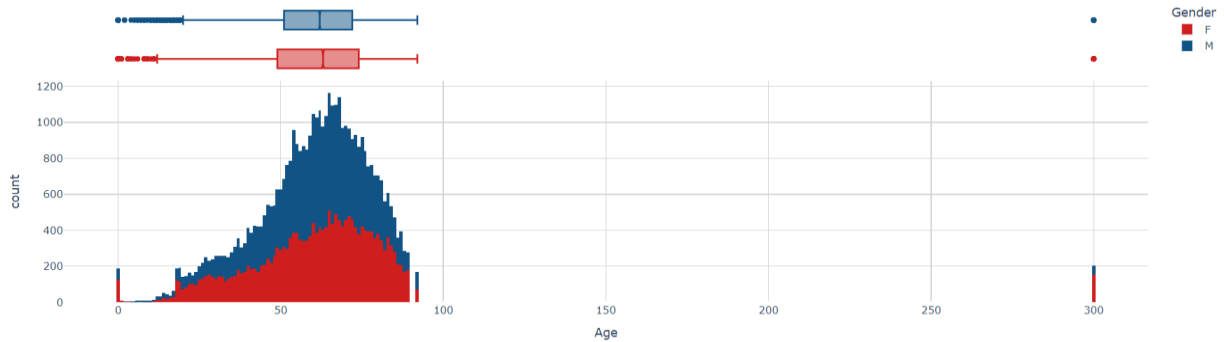


Figure 5.3: Age distribution per gender group and outlier detection

A key challenge in working with multi-institutional datasets is the variability in sampling rates. The PhysioNet dataset contains recordings sampled at three different frequencies:

- **500 Hz:** 42,511 recordings (98.63%)
- **1000 Hz:** 516 recordings (1.20%)
- **257 Hz:** 74 recordings (0.17%)

As can be noticed, most of the recordings (over 98%) are sampled at 500 Hz, which is a typical clinical standard. The remaining recordings, mostly from the PTB and St. Petersburg INCART datasets, are sampled at 1000 Hz and 257 Hz respectively. To ensure consistency across the dataset, all ECG signals were resampled to 500 Hz during preprocessing, as will be described in Chapter 7. This standardization step is necessary for the deep learning models to process the signals uniformly.

In terms of the classes studied in this collection, the dataset contains 111 unique diagnostic labels, reflecting the variety of cardiac conditions that can be identified through ECG signals. However, for this thesis, following the notion of the PhysioNet challenge, we also focus on a subset of 27 key diagnoses. These 27 classes were selected due to their clinical relevance and sufficient representation in the dataset. They include common arrhythmias, such as atrial fibrillation (AF) and atrial flutter (AFL), as well as various forms of heart block and bundle branch blocks. The selected classes are:

- 1st degree AV block (IAVB)
- Atrial fibrillation (AF)
- Atrial flutter (AFL)
- Bradycardia (Brady)
- Complete right bundle branch block (CRBBB)
- Incomplete right bundle branch block (IRBBB)
- Left anterior fascicular block (LAnFB)
- Left axis deviation (LAD)
- Left bundle branch block (LBBB)
- Low QRS voltages (LQRSV)
- Nonspecific intraventricular conduction disorder (NSIVCB)
- Pacing rhythm (PR)

- Premature atrial contraction (PAC)
- Premature ventricular contractions (PVC)
- Prolonged PR interval (LPR)
- Prolonged QT interval (LQT)
- Q wave abnormal (QAb)
- Right axis deviation (RAD)
- Right bundle branch block (RBBB)
- Sinus arrhythmia (SA)
- Sinus bradycardia (SB)
- Sinus rhythm (NSR)
- Sinus tachycardia (STach)
- Supraventricular premature beats (SVPB)
- T wave abnormal (TAb)
- T wave inversion (TInv)
- Ventricular premature beats (VPB)

Figure [5.4] illustrates the occurrence of each of the 111 diagnostic labels within the dataset. This distribution highlights the prevalence of conditions such as sinus rhythm (NSR), colored in yellow, which is expected due to its commonality. As this plot quantifies all the labels we can clearly see that there are a lot of labels that have almost 0 examples resulting to extra confusion for any potential model that will be trained using them.

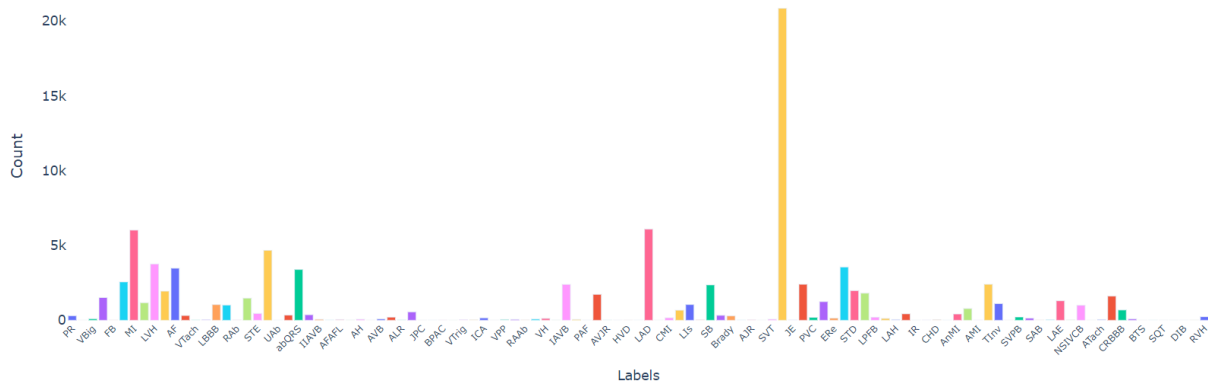


Figure 5.4: A countplot of all the 111 classes available in the dataset

Additionally, we focused on the distribution of the 27 key classes within the dataset. A second countplot (Figure [5.5]) shows the frequency of these selected classes. This subset emphasizes the clinical relevance of the model, as these conditions are among the most commonly diagnosed in clinical settings, making them highly valuable for real-world applications of wearable ECG monitoring. By focusing on the 27 classes we end up with patients that none of their diagnosed labels belongs to one of the selected classes and therefore they are excluded from the dataset. If we were to compare the previous distribution (Figure [5.4]) with that of Figure [5.5] we can infer that the balance of the data is significantly better but still there are some classes that are relatively undersampled.

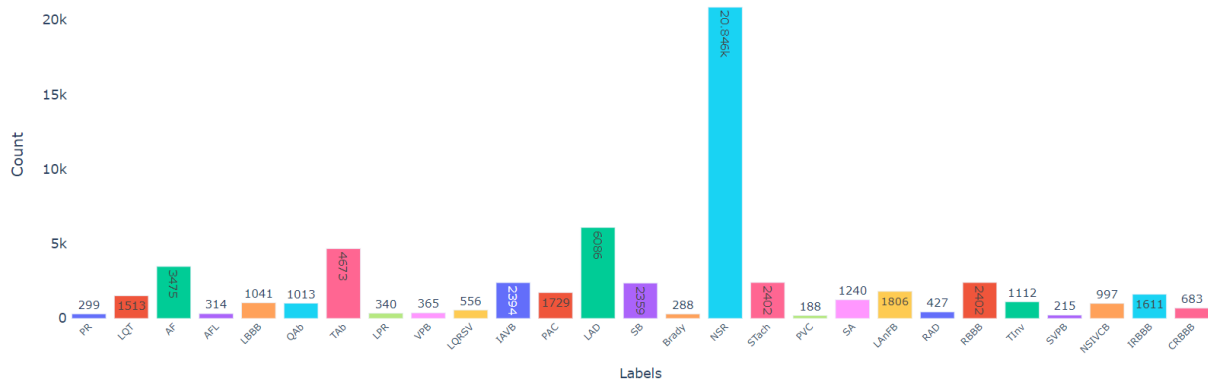


Figure 5.5: A countplot of all the 27 selected classes available in the dataset

As already mentioned, the PhysioNet dataset presents a multi-label classification problem, where each recording can be associated with multiple diagnostic labels. This characteristic reflects the complexity of ECG interpretation in clinical practice, where patients often exhibit more than one cardiac abnormality simultaneously. The number of co-occurring conditions varies across the dataset, with some patients having no diagnosis, while others may have up to seven conditions diagnosed from a single ECG recording as shown in Figure [5.6], with the majority of our records having one and two labels assigned to them.

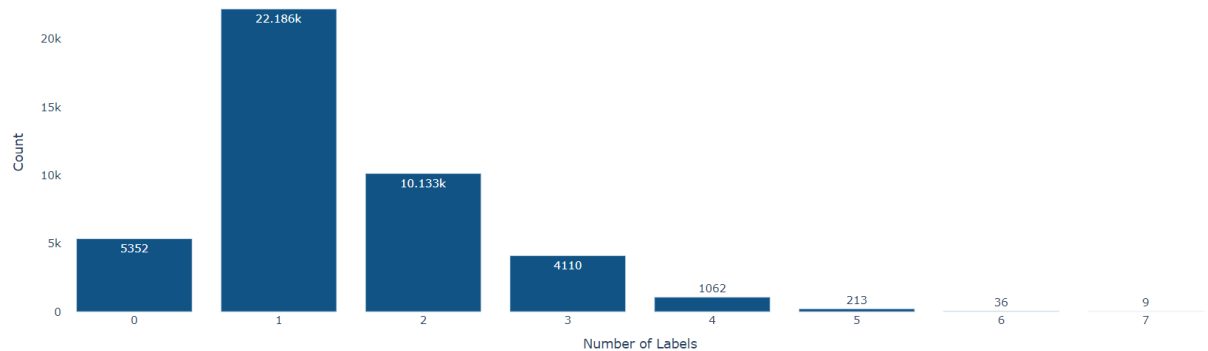


Figure 5.6: A countplot of the number of labels per ECG record

Finally, to ensure robust model evaluation and to mitigate the risk of overfitting, the dataset was split into training and testing sets. 20% of the dataset was reserved as a test set, while the remaining 80% was used for training and validation. For the training and validation phases, a stratified k-fold cross-validation approach was implemented. A thorough explanation of the training data methods used will be shared in Chapter 7.

To conclude, the PhysioNet dataset provides an extensive and diverse collection of ECG recordings, accompanied by rich demographic data and detailed diagnostic labels. This data forms the foundation for developing deep learning models capable of detecting multiple cardiac conditions in real-time, wearable devices. The selection of 27 key cardiac conditions for this study ensures that the models are clinically relevant and can generalize well across different populations. The multi-label nature of the dataset and its complex class distribution add

additional challenges, making this a highly valuable dataset for advancing the field of automated ECG analysis.

The variety in sampling rates, patient demographics, and the number of diagnoses per patient also ensure that the models are robust and can be applied in real-world scenarios. Future chapters will detail the preprocessing and model development techniques that were employed to transform this data into a useful tool for real-time arrhythmia detection.

6 Literature Review

Studies focusing on the automatic diagnosis of ECGs via ML and DL-based approaches have grown in popularity over the last years. Since the literature continues to grow interest in deep learning and its medical application it is only natural that the field is becoming also the center of attention for high-profile competitions such as the PhysioNet/Computing in Cardiology (CinC) Challenge [50]. In this event, a number of researchers have advanced the field of ECG classification through deep learning, by developing state-of-the-art algorithms that advance the boundary of automatic ECG analysis, combining new architectures with novel methodological developments. This chapter intends to focus on describing in depth works presented during the 2020 PhysioNet Challenge since it comprises the most relevant work to our dataset and thesis scope. Each of the contributions mentioned in this section brings novelty to address part of the challenge posed by ECG data, whether it be multi-label classification, imbalanced datasets, noisy signals, or domain generalization. It is worth mentioning that in the PhysioNet competition, the organizers used a held-out test set to internally evaluate each team's submitted projects. After the submission date, the evaluation process involved their own benchmarking system by creating a custom evaluation weighted metric, for which more details will be provided in the following chapters. By using the above-mentioned system, a leaderboard with the top ranked teams was created. Therefore, in this section, some of the highest in ranking as well as state-of-art approaches will be presented and compared thoroughly to make the readers comprehend how our methods differentiate with the already existing.

Natarajan et al. [54] proposed a wide and deep transformer neural network for their winning submission to the PhysioNet/CinC Challenge 2020. Their approach was notable for its combination of handcrafted features with deep learning representations, effectively blending traditional ECG analysis techniques with modern deep learning architectures. The “wide” part of their model relied on domain expertise for identifying clinically relevant features which included heart rate variability, QRS duration, and T-wave morphology. The handcrafted features that they engineered assisted the model with a solid foundation of known diagnostic indicators, helping to guide the learning process.

On the other hand, the “deep” part of their architecture utilized the power of the transformer architecture, which in very recent years has been proven to be groundbreaking in natural language processing and computer vision tasks. What makes Transformers excel at capturing long-range dependencies in sequential data is their self-attention mechanism, which allows the model to weigh the importance of each part of the input sequence dynamically. In the context of ECG diagnosis, the aforementioned has proven to be beneficial for identifying abnormalities that unfold over several heartbeats, such as atrial fibrillation or ventricular tachycardia.

One of the key innovations in Natarajan et al.'s model was its ability to process large datasets efficiently due to the parallelizable nature of the transformer architecture. Unlike traditional recurrent neural networks (RNNs) or long short-term memory (LSTM) networks, which process data sequentially, transformers can analyze entire sequences at once. This not only reduced

training time but also allowed the model to capture global patterns in the ECG data, which were critical for distinguishing between different types of arrhythmia. Their model achieved state-of-the-art performance, scoring highest in the challenge, and setting a new benchmark for ECG classification tasks.

A different but fascinating approach was Zhao et al. [55], who developed an adaptive ResNet-based model which featured Squeeze-and-Excitation blocks to enhance the explainability and performance in the ECG classification task. The proposed SE blocks enabled this network to adaptively change and assign the significance of each lead when processing the ECG, which is incredibly useful for 12-lead ECG records. As in real-world medical practices, a doctor would give more significance to some particular leads to make a diagnosis as some leads may carry more diagnostic importance than others, and the model of discussion was purposely designed to mimic this behavior by giving the more relevant leads greater weights in the training process.

As already mentioned, the architecture of the model was built on ResNet, which is a well-known deep learning architecture designed to mitigate the problem of vanishing gradients in deep networks. ResNet's skip connections allowed the model to preserve information from earlier layers, enabling it to learn deeper representations without suffering from the degradation of performance typically seen in very deep networks. What make their approach stand out even more is that Zhao et al. customized the ResNet architecture by introducing large kernel sizes in the convolutional layers, which helped the model capture long-term dependencies in the ECG signals. Taking this direction enabled them to detect conditions like bundle branch blocks, which affect the timing and shape of the QRS complex across multiple leads.

Finally, to address the issue of class imbalance, Zhao et al. employed a novel grid-search method to optimize class-specific thresholds during the training process. Moreover, this method ensured that the model did not overfit to more common classes, such as sinus rhythm, while neglecting rarer but clinically significant conditions like ventricular premature beats. The model's ability to adaptively weigh the ECG leads and dynamically adjust classification thresholds resulted in improved sensitivity and specificity, especially for challenging multi-label classification tasks, placing them one of the highest-ranking scores in the leaderboard.

Similarly to Zhao et al. , Zhu et al. [56], representing Team HeartBeats, introduced an ensemble model based on SE-ResNet to classify cardiac abnormalities from 12-lead ECG signals. Their model was placed third in the PhysioNet/CinC Challenge 2020, highlighting the effectiveness of combining deep learning with rule-based systems. For the same reasons as the previous team, they opted their approach to consist of Squeeze-and-Excitation (SE) blocks, which adaptively recalibrated the feature maps to assign higher weights to the most informative channels in the ECG signals leading their model to make more accurate diagnosis.

Specifically, the team employed an ensemble of SE-ResNet models, each trained on different segments of the ECG recordings. For example, one model was trained on 10-second segments, while another was trained on 30-second segments, allowing the ensemble to capture both short-

term and long-term dependencies in the ECG data. This approach ensured that the model could detect both transient arrhythmias, such as premature atrial contractions, and sustained conditions, like atrial fibrillation. What makes this work different from the others is the decision of the authors to introduce a rule-based model for bradycardia detection, which leveraged clinical knowledge about heart rate thresholds to improve classification accuracy. This hybrid approach of combining deep learning with clinical rules proved to be highly effective, particularly in reducing false negatives for certain classes.

Finally, Zhu et al. addressed the challenge of noisy and imbalanced data by customizing a multi-label loss function that emphasizes the cost of incorrect predictions. By penalizing the model more heavily for misclassifications in rare classes, they were able to achieve better balance across the 27 scored classes. Their results demonstrated the importance of integrating domain knowledge with advanced machine learning techniques, as the rule-based corrections provided significant improvements in clinical performance metrics.

Oppelt et al. [57] proposed a new hybrid model that incorporated a signal processing-based scatter transform into a deep residual neural network (ResNet). Their notion was to create a modulation of a wavelet transform aimed at capturing the small-scale morphological features of the time-series data of the P-wave or QRS complex in the ECG signal. Unlike traditional wavelet transforms, the scatter transform is non-trainable and derived from theoretical principles in such a way that the features it extracts are invariant against locally deforming the input signals.

Essentially, Oppelt et al. combined a scatter transform with a ResNet to take advantage of both theoretical feature extraction and trainable deep learning components. The ResNet in their method thus processed the scatter-transformed information, and the network was hence allowed to be focused on capturing more specific, higher-level representations relevant for ECG classification. The hybrid model therefore provides a unique balance between interpretability and flexibility, bringing stability and robustness to noise via a scattering transform while letting the ResNet capture some of the subtlety of the data not captured by the hand-engineered scattering transform alone.

Finally, one great remark of the work introduced by Oppelt et al. was that this approach worked particularly well for noisy signals: the application of scatter transformation at the first stage of data processing made the model much less sensitive to noise and baseline drift—the two most common types of ECG recording artifacts. The proof of their innovation is that their model reached the fourth-place score at the PhysioNet/CinC Challenge’s leaderboard and in turn underlined the efficiency of combining classic signal processing techniques with state-of-the-art deep learning approaches for time-series classification.

Hasani et al. [58] presented an adversarial multi-source domain generalization approach to cope with domain generalization in ECG classification. The approach was engineered to handle the diversity across ECG records emanating from a variety of different sets of hospitals, devices, and recording settings. This variability, also known as further domain shift, can reduce the

performance of machine learning models when those latter are used to predict test data deriving from a different distribution from their training set.

To tackle this issue, Hasani et al. proposed a model that combined CNNs with LSTMs for extracting features from the ECG signal; their convolutional neural network layers captured the spatial properties of ECG data, while the LSTM layers captured temporal dependencies, enabling this model to process information in space and time. The key innovation in their approach was the usage of adversarial domain generalization, which aimed to learn domain-invariant by training the model to discriminate between diverse sources while performing optimization over the classification objective.

In addition to the adversarial domain generalization framework, Hasani et al. employed extensive data augmentation techniques to simulate the effects of domain shift during training. These include adding noise, the random permutation of leads, and various kinds of filters applied to the ECG, hence enhancing the robustness of the model coming from different domains. Their approach ranked fifth in the PhysioNet/CinC Challenge, hence solidifying the importance of domain shift overcoming in ECG multisource datasets.

7 Data Preprocessing

Preprocessing is without a doubt one of the most fundamental parts of every machine learning pipeline, and especially relevant in physiological inputs such as electrocardiograms. The raw ECG signals are always susceptible to a lot of noise, artifact, and other inconsistencies attributed to various factors related to the movement of patients, electrode positioning, and recording apparatus. If not treated accordingly, these might substantially bring down the performance of deep learning models. In this chapter, we present the preprocessing pipeline that normalizes, cleans, and segments the ECG data in a manner useful for both model training and model inference. The specific preprocessing techniques discussed within the thesis include filtering, resampling, and normalization, in addition to the feature extraction from demographic information such as age and gender.

7.1 Overview of the Preprocessing Pipeline

The whole preparation workflow described in this chapter is encapsulated within the special PyTorch dataset class, `PhysioNet_Dataset`, and its subroutines for loading ECG data from source `.mat` files and doing several preprocessing steps. Preprocessing is done in order to standardize the data arriving from different recordings and to ensure that our training models will receive a standardized size of input data. Besides ECG signal processing, demographic features (age and gender) are also processed to serve as additional inputs for the model.

The pipeline takes as input a list of unique patient IDs and a `DataFrame` containing metadata for each recording, including the file paths, sampling rates, and labels. The following sections provide a detailed explanation of each preprocessing step, as well as its role in preparing the data for training deep learning models.

7.2 Data Loading and Resampling

The first step in the preprocessing pipeline is to load the raw ECG data from `.mat` files. The ECG recordings are sampled at different frequencies depending on the source institution, which can range from 250 Hz to 1 kHz. To ensure uniformity, all recordings are resampled to a standard frequency of 500 Hz, which is commonly used in clinical ECG analysis. This step is important because deep learning algorithms are very sensitive to temporal anomalies in time-series data. Uniform sampling enables the algorithm to catch the patterns, which do not depend on specific recording conditions.

The resampling is performed using either decimation or interpolation techniques, depending on whether the original sampling rate is higher or lower than 500 Hz. Decimation reduces the sampling rate by discarding some data points, while interpolation fills in missing data points to increase the sampling rate. The described step acts as a necessary normalization of all ECG records to one standard temporal resolution, regardless of originally used recording settings. ECG signals, after resampling, are stored as NumPy arrays for subsequent processing.

7.3 Filtering

Once the ECG data has been loaded and resampled, the next step is to apply a bandpass filter to remove noise and artifacts. ECG signals are often contaminated by various sources of noise, such as baseline wander (low-frequency noise caused by patient movement) and powerline interference (high-frequency noise at 50 or 60 Hz). The filtering process is designed to isolate the frequency range of interest for ECG analysis, which typically lies between 3 Hz and 45 Hz. This range captures most of the clinically relevant features of the ECG signal, such as the P-wave, QRS complex, and T-wave, while excluding extraneous noise.

Moreover, a Finite Impulse Response (FIR) bandpass filter [59] is applied to the ECG signals to remove frequencies outside of a given range. FIR filters are widely used in ECG preprocessing since they offer linear phase characteristics, i.e. they do not distort the timing relationships between different components of the ECG waveform. The implementation of the filter is done using an existing python module, which allows for easy configuration of the filter order and cutoff frequencies. To ensure a sharp transition between the passband and stopband and at the same time maintain computational efficiency, the filter order is set to 0.3 times the sampling frequency as proposed by the literature.

7.4 Signal Normalization

After filtering, the ECG signals are then normalized to ensure a standardized range of amplitudes. Normalization is considered a particularly important preprocessing step for training deep learning models because it prevents biases caused by large variations in signal amplitude among the different recordings. Each ECG recording in normalization will be scaled to zero mean and unit variance, hence normalizing the dataset's signal amplitudes. This step ensures that the model focuses on the structure alone of the ECG waveforms regardless of any variation in signal amplitude.

Besides regular normalization, we also use smoothing scaling technique that is introduced aiming to avoid division by zero during the regularization process. The scaling factor is modified by adding a small constant termed "smooth" inside the denominator. The formula used in this step is as follows:

$$normalized\ signal = 2 \times \frac{(signal - \min(signal))}{(\max(signal) - \min(signal) + smooth)} - 1$$

After a signal pass through this formula, it scales its' values to a range between -1 and 1, ensuring that all recordings are harmonized.

7.5 Random Window Extraction

As one can imagine, ECG recordings vary in length, and usually the longer the recordings the larger the information that they contain about the patient's heart activity. To standardize the

input size for the model, a fixed-length window of 10.24 seconds is extracted from each ECG recording. We chose the specific window size based on the duration of the typical ECG beat sequence, allowing the model to capture multiple heartbeats within each window. For recordings that are longer than the window size, random windows are sampled during training to provide a diverse range of segments to the models that we will train. This random windowing approach also acts as a form of data augmentation, as the model is exposed to different parts of the ECG signal in each training epoch making this step a not only a preprocessing step but also a model performance regularization one.

In this step of our preprocessing pipeline, for recordings that are shorter than the window size, the signals are padded with zeros to match the required length. Zero-padding is a common technique used in time-series analysis to ensure that all inputs to the model have the same dimensionality. The windowing and padding operations are crucial for ensuring that the model can process recordings of varying lengths while maintaining a consistent input size.

7.6 Demographic Feature Processing

As will be discussed in the next chapter, one of our experiments on this thesis is the Wide and Deep architecture, which will utilize tabular information on the Wide part. The respective demographic information used are the patients' age and gender along their ECG. Therefore, as part of our preprocessing we first extract the age and any outlier ages shown in the previous chapter are removed from the dataset (e.g. age samples with a value of 300). For the gender variable we first standardize the values to common naming conventions as it was shown that some sources were referring to each gender by its full word and others by its initial (e.g. "Female" and "F"). After this harmonization, one-hot encoding for gender: "male" = 1, "female" = 0 was applied. The ambition of bringing this type of information into the model is to color our model more, as some heart diseases can affect people of particular age brackets or even a specific gender.

7.7 Target Variable Handling

Finally, as our focus is shifted into the 27 classes mentioned above we standardize the order that the labels are sorted for all the patients and we encode them into a binary format resulting into a 27 element long vector of zeros and ones. It is worth mentioning that patients that didn't have any of the 27 labels of interest in their diagnoses were excluded from the dataset.

This chapter described the necessary preprocessing steps for preparing the ECG recordings to be used in a deep learning model, which include resampling, filtering, normalization, window extraction, and demographic feature processing-very crucial preprocessing steps to make them clean, coherent, and suitable for model training. The step of cleaning and preprocessing of data is very important for high accuracy and robustness in models, especially on complex multi-label classification issues. In the following charts we can visualize how a raw ECG versus a preprocessed ECG looks like. In Fig [7.1] we see the raw ECG of a male patient that is 70 years

old with a sample rate of 500 Hz and how his ECG is transformed after the preprocessing pipeline (Fig [7.2]).

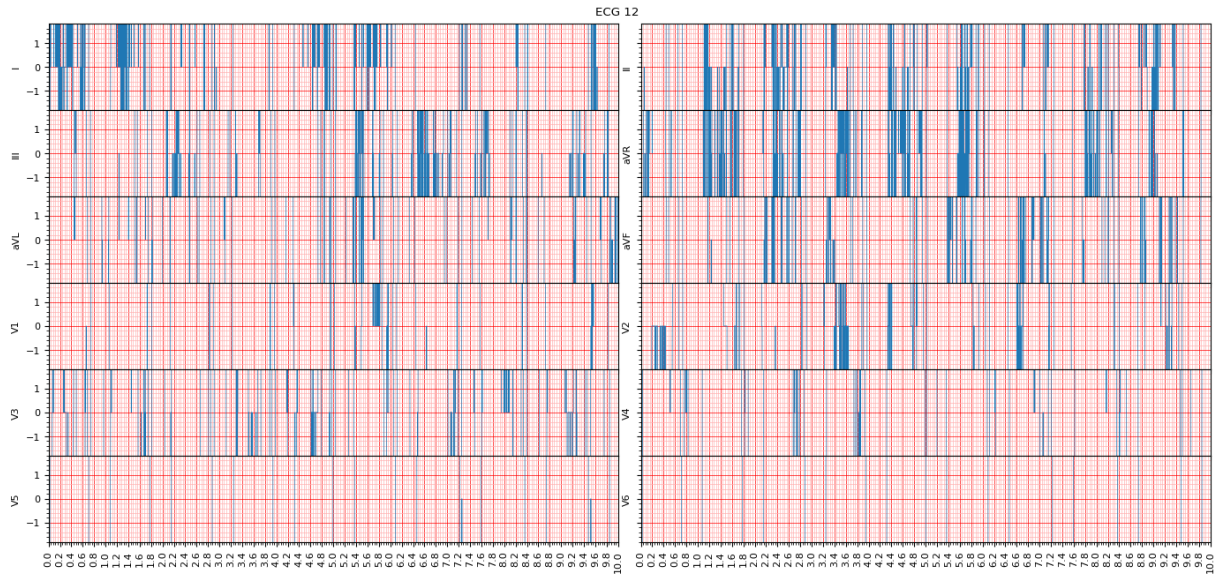


Figure 7.1: Visual representation of raw 12-lead ECG record from our training set

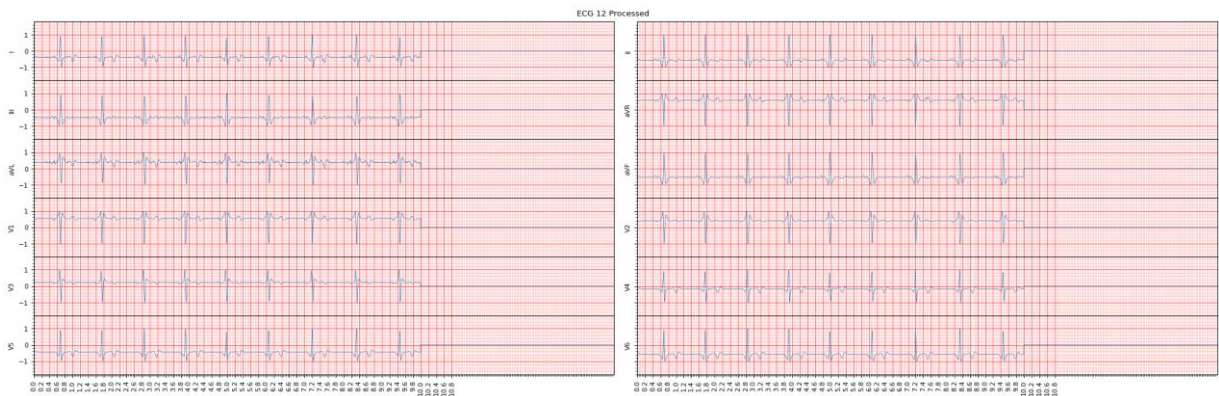


Figure 7.2: Visual representation of raw 12-lead ECG record from our training set after preprocessing

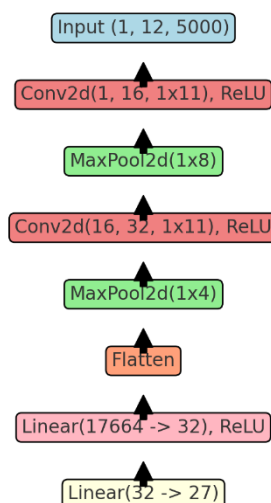
8 Methods

This chapter examines the architectures and methodology utilized in this thesis for the classification of arrhythmias from electrocardiogram (ECG) data. The subsequent sections delineate four deep learning architectures trained, starting with a basic CNN model utilized as a reference point. Each model was engineered with distinct characteristics to tackle the intricacies of the ECG signal, the multi-label classification challenge, and the limitations necessitated by the requirement for lightweight, real-time implementation in wearable devices. It is reminded that the models discussed in these sections are trained using a 5 k-fold stratified cross validation and the results on each model concern only the fold of which the model outperformed the rest of the folds.

8.1 Baseline Convolutional Neural Network (CNN) Architecture

For this study, the SimpleECGConvNet model was employed as a baseline architecture for ECG classification tasks. The network's architecture was specifically designed to balance simplicity with performance, minimizing the number of parameters to ensure efficient training while maintaining sufficient depth to capture complex patterns in the ECG data. The model consists of two convolutional layers, each followed by a ReLU activation function and max-pooling layers. The first convolutional layer has 16 filters, each with a kernel size of 1×11 and stride of 5, followed by a max-pooling operation with a pool size of 1×8 . The second convolutional layer has 32 filters with the same kernel size and padding, followed by a max-pooling operation with a smaller pool size of 1×4 . The output of the second convolutional layer is flattened, and a fully connected layer with 32 neurons is applied, followed by a ReLU activation function and an output layer with 27 neurons corresponding to the target classes.

SimpleECGConvNet Model Architecture



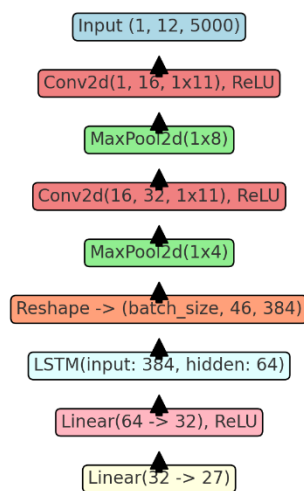
The total number of parameters in this architecture was 313,659. The model was trained using a batch size of 256 over a maximum of 100 epochs, with early stopping enabled if the validation

loss did not improve after 2 epochs. The most actual epochs trained during the folds were 27. The Adam optimizer was employed with a learning rate of 0.001, and binary cross-entropy with logits served as the loss function. The training set consisted of 30,236 examples, while the validation set contained 4,335 examples. On the final epoch, the model achieved a training loss of 0.101, accuracy of 20.5%, and an F1-macro score of 0.3602. The corresponding validation metrics were a loss of 0.125, accuracy of 15%, and an F1-macro score of 0.31.

8.2 Convolutional and Long Short-Term Memory (ECGConvLSTMNet) Architecture

The ECGConvLSTMNet architecture was designed to leverage both convolutional and recurrent layers to capture both spatial and temporal dependencies in the ECG data. The model starts with two convolutional layers, followed by max-pooling layers, similar to the previous models. The first convolutional layer contains 16 filters with a kernel size of 1×11 and a stride of 5, while the second contains 32 filters with the same kernel size. The output of the second convolutional layer is reshaped and fed into an LSTM layer, which captures the temporal dependencies across the ECG signals. The LSTM has 64 hidden units and processes sequences of length 46 with an input size of 384 (corresponding to the flattened output of the convolutional layers). The final output is obtained through two fully connected layers.

ECGConvLSTMNet Model Architecture

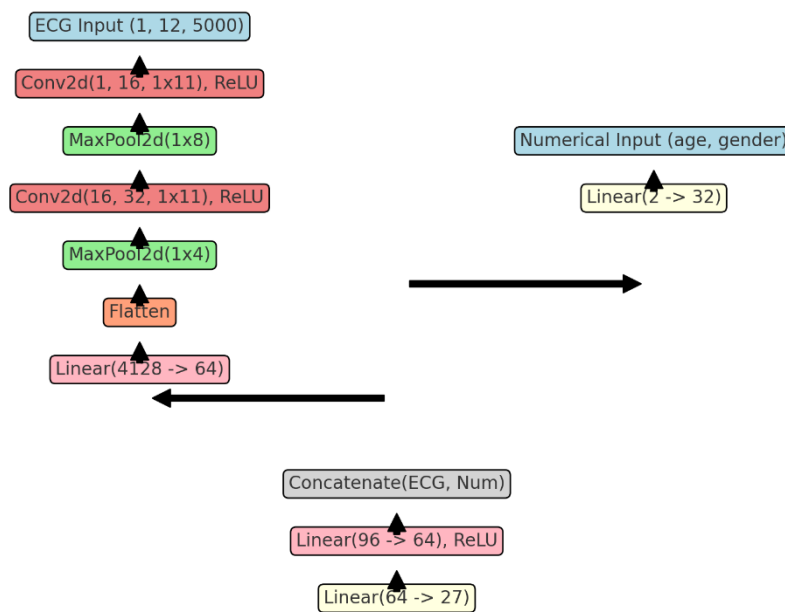


This model has 124,027 parameters and was trained using a batch size of 256 over 100 epochs, with early stopping set to trigger if the validation loss did not improve for 2 epochs. The model trained for 35 epochs in the longest fold. The Adam optimizer was used with a learning rate of 0.001, and binary cross-entropy with logits was applied as the loss function. The training dataset included 30,236 examples, while the validation dataset contained 4,335 examples. At the end of training, the model achieved a training loss of 0.123, accuracy of 31.54%, and an F1-macro score of 0.26. The validation metrics indicated a loss of 0.1588, accuracy of 31.8%, and an F1-macro score of 0.264.

8.3 Wide and Deep Neural Network (WideAndDeepECGNet)

The WideAndDeepECGNet is a hybrid model designed to integrate both ECG signal data and additional demographic information such as age and gender. This architecture consists of two separate input paths: the wide path for numerical features and the deep path for the ECG data. The deep path processes the ECG data through two convolutional layers, each followed by max-pooling. The first convolutional layer uses 16 filters with a kernel size of 1×11 , while the second uses 32 filters. The output of the convolutional layers is flattened and passed through a fully connected layer with 64 neurons. The wide path processes the numerical input (age and gender) through a fully connected layer with 32 neurons. The outputs of the two paths are concatenated and passed through a final fully connected layer to produce the final output.

WideAndDeepECGNet Model Architecture (Dual Input)



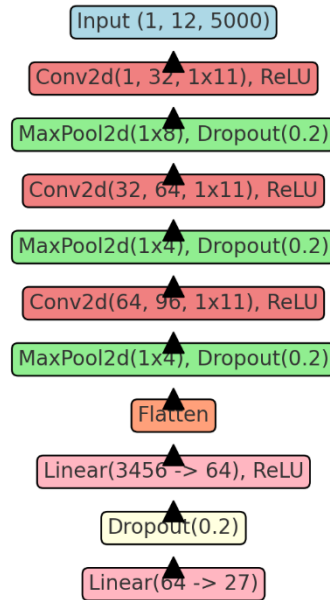
This architecture contains 278,171 parameters and was trained over 100 epochs with a patience of 2 for early stopping. The most epochs trained in a fold were 28. The training and validation sets consisted of 30,236 and 4,335 examples, respectively. The Adam optimizer was used with a learning rate of 0.001, and binary cross-entropy with logits was the chosen loss function. In the final epoch, the model achieved a training loss of 0.22, accuracy of 26.7%, and an F1-macro score of 0.11. The validation set yielded a loss of 0.1858, accuracy of 28%, and an F1-macro score of 0.13.

8.4 Enhanced Convolutional Neural Network (ECGConvNet)

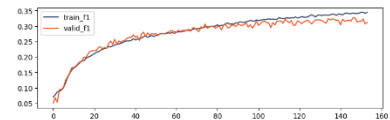
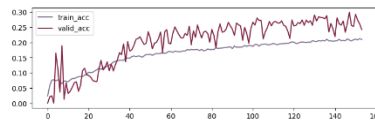
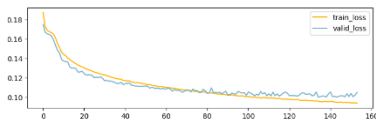
The ECGConvNet model builds upon the baseline architecture by incorporating additional layers to enhance the model's ability to capture more intricate patterns in the ECG data. This model consists of three convolutional layers, each followed by ReLU activations, max-pooling, and dropout layers to prevent overfitting. The first convolutional layer uses 32 filters with a

kernel size of 1×11 and a stride of 5, while the subsequent layers contain 64 and 96 filters, respectively. These layers are followed by fully connected layers to produce the final classification.

ECGConvNet Model Architecture with Layer Details



The model comprises a total of 572,027 parameters and was trained over a maximum of 1000 epochs with a patience of 10 for early stopping. The model was trained on a batch size of 128, and the Adam optimizer was used with a learning rate of 0.001. The loss function was binary cross-entropy with logits. During training, the model was provided with 30,236 examples for training and 4,335 for validation. After 144 epochs, the model achieved a training loss of 0.0983, an accuracy of 32%, and an F1-macro score of 0.33. On the validation set, the loss was 0.1124, with an accuracy of 21% and an F1-macro score of 0.326, indicating the model's reasonable performance without significant overfitting.



8.5 Quantized model (QECGConvNet)

Based on the validation results we can conclude that the best performance was that of ECGConvNet, outperforming all the other architectures proposed in this chapter. Therefore, from this chapter and onwards we select this particular model for further usage and evaluation on the test set. Building upon the theoretical concepts of quantization discussed earlier, we

applied post-training quantization to the best-performing ECGConvNet model in this thesis. Post-training quantization, as described, is one of the most accessible techniques for transforming a fully trained model into a lighter version by reducing the precision of its weights and activations from 32-bit floating-point representations to more computationally efficient 8-bit integers. This approach significantly reduces memory usage and speeds up computations, making the model suitable for deployment on wearable devices where real-time predictions are essential.

In the context of this work, the quantization process followed a structured four-step procedure to convert the ECGConvNet model into its quantized counterpart, QECGConvNet. Initially, we performed layer fusion, combining convolutional and ReLU activation layers into single operations. This step helps streamline the computation and reduce overhead, which is particularly important for resource-constrained environments like wearable devices. Moreover, the model was prepared for static quantization using the QNNPACK backend, optimized for ARM architectures common in wearables. Calibration was then performed using a subset of the training data to adjust the activations and ensure that the transition to lower precision did not significantly impact the model's performance.

As noted in the theoretical chapter, while post-training quantization can introduce some degradation in model accuracy, it remains one of the most practical methods when the model needs to be deployed on devices with limited computational resources. The choice of post-training quantization here aligned with the thesis's goal to create a lightweight model suitable for wearable applications, without requiring the computational intensity of quantization-aware training (QAT). The calibration step was crucial in mitigating the accuracy loss, ensuring that the quantized model could still perform diagnostic tasks effectively despite the reduction in precision.

The resulting QECGConvNet demonstrated the advantages of post-training quantization, achieving significant reductions in memory and computational cost, while hopefully maintaining a performance level close to the original model.

8.6 Individual probability threshold tuning

To further optimize the performance of both the ECGConvNet and QECGConvNet models, we employed a custom validation generator and fine-tuned the probability threshold for each of the 27 output classes. In multi-label classification tasks, a common approach is to apply a default threshold (usually 0.5) to convert predicted probabilities into binary labels. However, different classes may benefit from different thresholds to maximize performance metrics like F1-macro. By fine-tuning the probability threshold for each class individually, we aimed to improve the macro-averaged F1 score, which is particularly sensitive to the imbalances inherent in the multi-label ECG dataset.

For this process, we evaluated thresholds ranging from 0.05 to 0.65 in increments of 0.05. For each fold in the cross-validation process, we calculated several metrics, including accuracy, F1-

macro score, and the challenge metric, across all threshold values. The model was then evaluated with the optimal threshold settings that produced the highest F1-macro score for each class. This tuning was performed on both the ECGConvNet and its quantized counterpart, QECGConvNet, ensuring that the models were not only computationally efficient but also well-calibrated to produce optimal diagnostic performance.

The fine-tuning process leveraged the outputs from the validation set, adjusting the predicted probabilities to the most appropriate threshold for each individual diagnosis. By doing so, we were able to minimize the loss in accuracy that can occur when a single threshold is applied uniformly across all classes, especially in the case of imbalanced datasets like the one used in this thesis. This class-specific thresholding allowed for a more nuanced and effective classification of the 27 cardiac conditions, thereby aligning the model's predictions more closely with the clinical requirements for wearable ECG diagnostics.

8.7 Evaluation Results

As already mentioned, the evaluation of the models in this thesis is based on a custom metric of the PhysioNet Challenge [50] designed to account for the complexity of multi-class, multi-label classification, particularly in the context of clinical diagnosis. Unlike traditional evaluation metrics, which typically assign equal penalties to all misclassifications, this custom metric incorporates clinical relevance by weighing different types of classification errors according to their clinical impact. Specifically, the scoring system begins with a multi-class confusion matrix $A=[a_{ij}]$, which tracks how frequently a classifier predicts each class relative to the true class across a set of recordings. For a given recording k , the matrix entry a_{ij} is updated if the true label c_i and predicted class c_j are both positive. This contribution is normalized by the number of positive labels and/or classifier outputs for the recording, represented as $|x_k \cup y_k|$, where x_k and y_k are the sets of true and predicted classes, respectively. The confusion matrix is computed as:

$$a_{ij} = \sum_{k=1}^n a_{ijk}, \text{ where } a_{ijk} = \frac{1}{|x_k \cup y_k|}, \text{ if } c_i \in x_k, \text{ and } c_j \in y_k$$

Next, a reward matrix $W=[w_{ij}]$ is applied to the confusion matrix. This matrix, designed by cardiologists, assigns full credit to correct classifications and partial credit to clinically similar misclassifications and is illustrated in Fig [8.7]. The unnormalized score s_U is calculated as the weighted sum of the confusion matrix entries:

$$s_U = \sum_{i=1}^m \sum_{j=1}^m w_{ij} a_{ij}$$

where m represents the total number of classes. To make the score more interpretable, it is normalized to fall between 0 and 1. This is done by comparing the observed score to two baselines: a classifier that always predicts the correct labels (yielding a score s_T) and an inactive classifier that always predicts the normal class (yielding a score s_I). The normalized score s_N is calculated as:

$$S_N = \frac{s_U - s_I}{s_T - s_I}$$

where s_T represents the score of a perfect classifier, and s_I represents the score of the inactive classifier. This normalization ensures that a perfect classifier achieves a score of 1, while an inactive classifier achieves a score of 0.

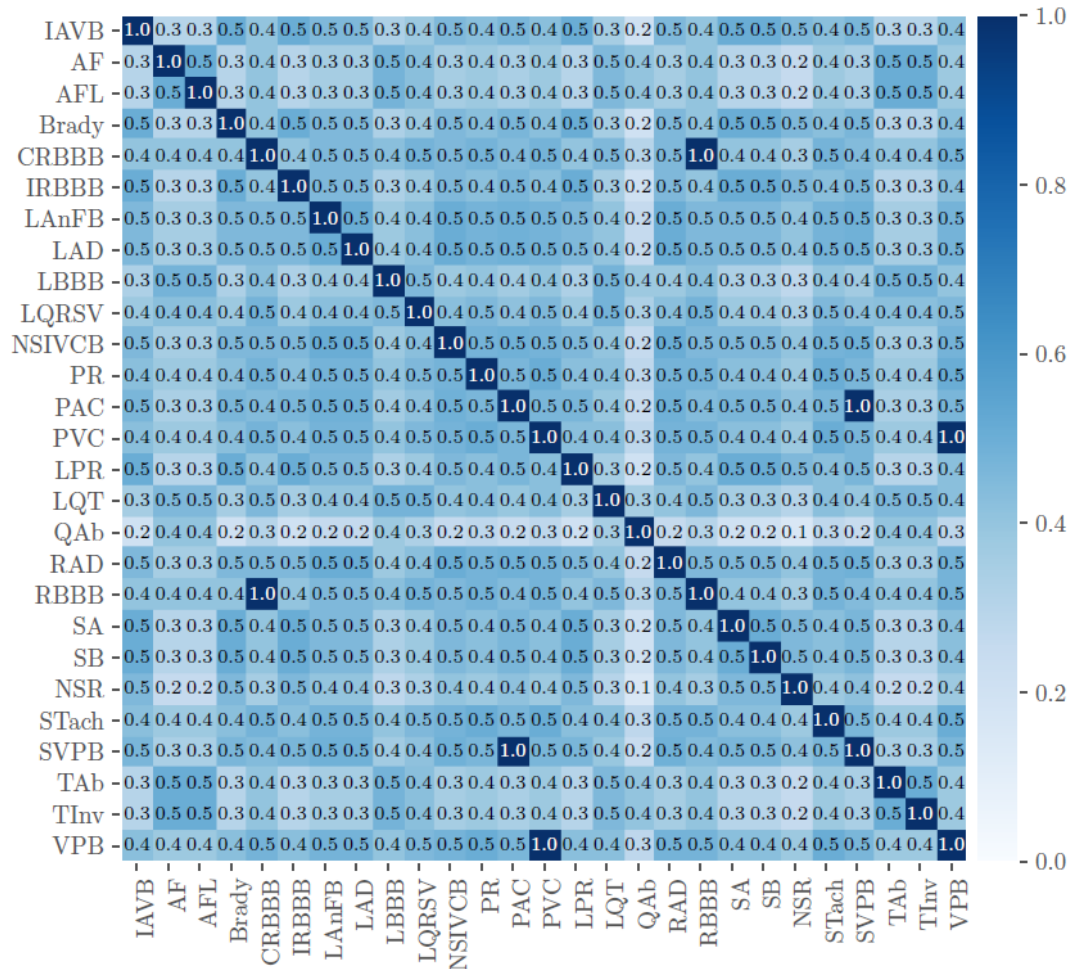


Figure 8.7: Reward matrix proposed by the challenge [50]

The evaluation of both the original ECGConvNet model and its quantized version, QECGConvNet, was conducted using the custom metric described above on the test set that we kept held-out. The evaluation results of our 2 selected models based on the challenge metric are being shown in Table 8.7 along with the corresponding metric of the researchers discussed in the literature review of the thesis.

| Model | Challenge Metric |
|-----------------------|------------------|
| Natarajan et al. [54] | 0.53 |
| Zhao et al. [55] | 0.52 |
| Zhu et al. [56] | 0.51 |
| Oppelt et al. [57] | 0.49 |
| ECGConvNet | 0.46 |
| Hasani et al. [58] | 0.44 |
| QECGConvNet | 0.39 |

Table 8.7: Evaluation results compared to the literature

The evaluation of the original ECGConvNet and its quantized counterpart, QECGConvNet, demonstrates the trade-off between model performance and computational efficiency, as reflected by the custom challenge metric. As seen in Table [8.7], the ECGConvNet achieved a score of 0.46, which is competitive compared to the state-of-the-art models by the literature. The quantized version, QECGConvNet, while still achieving a respectable score of 0.39, exhibits a slight reduction in performance due to the post-training quantization process, which is expected given the precision loss when reducing the model's complexity. Nevertheless, the quantized model's performance remains within a reasonable range, underscoring its potential for deployment in resource-constrained environments, such as wearable devices, where computational efficiency is paramount. This trade-off between performance and efficiency highlights the effectiveness of quantization techniques in maintaining diagnostic utility while meeting practical deployment requirements. Finally, it should be noted for comparison purposes that the ECGConvNet model that we trained is much lighter than the models proposed in the literature (shown in Table [8.7]); it contains 572,027 parameters while the Natarajan et al.'s [54] proposed model consists of 13.644 million parameters making it about 23.9 times larger.

9 Conclusions

This work aims to investigate whether it is possible to create a Deep Learning model for arrhythmia prediction that could then be quantized into a lightweight model capable of deployment onto wearable devices. In general, the goal is to keep the model performing properly in real-time predictions while maintaining the strict hardware limitations of wearables. This has also followed a structured path toward its fulfillment by reviewing the literature, understanding the problem, preprocessing the ECG data, and exploring a number of deep learning architectures and methods that may be used for improving the classification of ECGs. We reviewed the literature of various research related to the detection of arrhythmias using deep learning models, including state-of-the-art trends in wearable technology for real-time medical diagnostics. Our primary data source was the PhysioNet/Computing in Cardiology Challenge 2020 dataset which included 12-lead ECG recordings in order to classify 27 specified cardiac abnormalities. Considering the sheer number of recordings that required normalization, signal filtering, and resampling, a lot of preprocessing was done on the data.

The conducted experiments covered several different architectures, which were developed using both convolutional and recurrent neural network layers in the interest of modeling both spatial and temporal dependencies of ECG signals, including two state-of-the-art models: ECGConvNet and ECGConvLSTMNet. Also, a tailored metric of the PhysioNet scoring system that better reflects the real-world needs of arrhythmia classification as it has been given by domain experts was used to evaluate the models in a literature compatible manner.

Finally, we employed model quantization techniques to reduce the size of the trained models, making them compatible with the limited processing power of wearable devices. Specifically, post-training quantization was used to convert the model's parameters to lower precision formats, such as 8-bit integers, without significantly compromising accuracy.

9.1 Key Findings

The final quantized model achieved promising results but also demonstrates room for improvement. The baseline model (ECGConvNet) reached an F1-macro score of 0.264 on the validation set, while the quantized version maintained an acceptable performance but with a degradation in accuracy due to the inherent limitations of post-training quantization. Despite this, the quantized model is a promising candidate for future optimization, and further exploration of advanced quantization techniques, such as quantization-aware training or model distillation, could yield better results.

The overall findings suggest that the proposed approach—developing deep learning models for arrhythmia detection and subsequently quantizing them for wearable devices—is feasible. However, achieving state-of-the-art performance in real-time applications will require further work, especially in improving quantization techniques and exploring more advanced model architectures like transformers.

9.2 Limitations

Several limitations were encountered during the research process. To begin with, resource constraints played a significant role, as the models had to be trained mostly on CPU based machines, leading to long training times (approximately 10 minutes per epoch). The limited computational resources also restricted the number of epochs we could run, which could have impacted the final performance of the models. Secondly, the lack of domain expertise held us back in terms of the feature engineering, preprocessing techniques and creativity in the architectures used for the complex ECG data. With greater access to domain knowledge, particularly in cardiology, it might have been possible to come up with more custom architectures and models more effective on clinical applications. Additionally, the imbalanced nature of the dataset, with some arrhythmia classes significantly underrepresented, made it difficult to achieve high performance across all classes. Although techniques such as threshold tuning were employed to address this imbalance, further work is needed to handle rare classes more effectively.

9.3 Future Work

The results of this thesis provide a foundation for future research into lightweight models for wearable ECG monitoring, but there are several areas for further investigation. Future work could benefit from exploring more advanced model architectures, such as the transformer one, which has demonstrated superior performance in handling sequential data like ECG signals as shown in chapter 6. Feature engineering techniques that are incorporating domain expertise, maybe in conjunction with clinicians, could further enhance model performance by highlighting the most relevant features produced by the ECG data. Moreover, additional quantization techniques, such as model distillation and quantization-aware training, should be explored to minimize the performance degradation seen in the post-training quantization phase. These methods could yield models that maintain higher accuracy while remaining lightweight enough for wearable devices. Finally, it would be valuable to incorporate more real-world constraints into the model development process, such as battery consumption and processing power limitations, to ensure that the models are not only accurate but also practical for continuous use in wearable devices.

In conclusion, this thesis has demonstrated the feasibility of developing lightweight models for arrhythmia detection that can be deployed on wearable devices. While the final quantized model shows some promise, significant improvements can still be made, particularly through the exploration of more advanced architectures, feature engineering, and quantization techniques. With the continued advancement of deep learning and hardware technologies, it is likely that future developments will result in state-of-the-art arrhythmia detection systems capable of real-time operation on wearable devices. This could significantly impact the early diagnosis and management of cardiovascular diseases, potentially saving lives.

Bibliography – References – Online sources

- [1] World Health Organization, "Cardiovascular diseases (CVDs) fact sheet," 2021. [Online]. Available: [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)).
- [2] E. G. Nabel and E. Braunwald, "A tale of coronary artery disease and myocardial infarction," *The New England Journal of Medicine*, vol. 366, no. 1, pp. 54–63, 2012. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4357310/>.
- [3] G. A. Roth et al., "Global, regional, and national burden of cardiovascular diseases for 10 causes, 1990 to 2015," *The Lancet*, vol. 388, no. 10054, pp. 2561–2624, 2018. [Online]. Available: [https://www.thelancet.com/journals/lancet/article/PIIS0140-6736\(18\)31094-5/fulltext](https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(18)31094-5/fulltext).
- [4] S. Yusuf et al., "Global burden of cardiovascular diseases and risk factors, 1990–2020," *The New England Journal of Medicine*, vol. 382, no. 10, pp. 873–887, 2020. [Online]. Available: <https://www.nejm.org/doi/full/10.1056/NEJMra1900500>.
- [5] M. Nichols et al., "European cardiovascular disease statistics 2014," *European Heart Journal*, vol. 35, no. 42, pp. 2950–2959, 2014. [Online]. Available: <https://academic.oup.com/eurheartj/article/35/42/2950/407693>.
- [6] P. M. Rautaharju et al., "AHA/ACCF/HRS recommendations for the standardization and interpretation of the electrocardiogram: Part IV: The ST segment, T and U waves, and the QT interval," *European Heart Journal*, vol. 30, no. 6, pp. 673–679, 2009. [Online]. Available: <https://academic.oup.com/ehj/article/30/6/673/2887411>.
- [7] X. Jouven et al., "The Electrocardiogram in Cardiovascular Risk Prediction," *Circulation Research*, vol. 116, no. 12, pp. 2120–2130, 2015. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4601345/>.
- [8] A. Saxena et al., "Preventing cardiovascular disease in the developing world: the role of electrocardiography in subclinical detection of risk," *Bulletin of the World Health Organization*, vol. 96, no. 7, pp. 470–477, 2018. [Online]. Available: <https://www.who.int/bulletin/volumes/96/7/17-198846/en/>.
- [9] P. W. Macfarlane et al., "Electrocardiography: past and present," *Clinical Research in Cardiology*, vol. 99, no. 9, pp. 573–581, 2010. [Online]. Available: <https://link.springer.com/article/10.1007/s00392-010-0218-8>.
- [10] G. Hindricks et al., "2021 ESC Guidelines for the diagnosis and treatment of atrial fibrillation developed in collaboration with the European Association for Cardio-Thoracic Surgery (EACTS)," *European Heart Journal*, vol. 42, no. 5, pp. 373–498, 2021. [Online]. Available: <https://academic.oup.com/europace/article/23/5/850/6159470>.

- [11] C. T. January et al., "2019 AHA/ACC/HRS focused update on atrial fibrillation: A report of the American College of Cardiology/American Heart Association Task Force," *Circulation*, vol. 140, no. 2, pp. e125–e151, 2019. [Online]. Available: <https://www.ahajournals.org/doi/full/10.1161/CIR.0000000000000659>.
- [12] J. W. Mason et al., "Electrocardiographic reference ranges derived from 79,743 ambulatory subjects," *Journal of Electrocardiology*, vol. 40, no. 3, pp. 228–234, 2007. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1955230/>.
- [13] G. Montalescot et al., "Pretreatment with prasugrel in non–ST-segment elevation acute coronary syndromes," *The New England Journal of Medicine*, vol. 369, no. 11, pp. 999–1010, 2013. [Online]. Available: <https://www.nejm.org/doi/full/10.1056/NEJMra1110107>.
- [14] A. L. Goldberger, "Electrocardiography," *European Heart Journal*, vol. 27, no. 10, pp. 1162–1169, 2006. [Online]. Available: <https://academic.oup.com/ehj/article/27/10/1162/2844449>.
- [15] B. Surawicz and T. K. Knilans, *Chou's Electrocardiography in Clinical Practice: Adult and Pediatric*, 6th ed. Philadelphia: Saunders, 2008. [Online]. Available: <https://academic.oup.com/europace/article/12/5/707/519071>.
- [16] P. Kligfield et al., "Recommendations for the standardization and interpretation of the electrocardiogram: Part I," *European Heart Journal*, vol. 30, no. 4, pp. 529–536, 2007. [Online]. Available: <https://academic.oup.com/ehj/article/30/4/529/2884737>.
- [17] D. Noble, "Myocardial ischemia and infarction: Clinical insights," *European Heart Journal*, vol. 27, no. 6, pp. 1416–1424, 2006. [Online]. Available: <https://academic.oup.com/ehj/article/27/6/1416/2396320>.
- [18] B. Surawicz, "The role of electrocardiography in clinical decision-making," *European Heart Journal*, vol. 28, no. 8, pp. 985–993, 2006. [Online]. Available: <https://academic.oup.com/ehj/article/28/8/985/461719>.
- [19] A. Bayés de Luna et al., "Current electrocardiographic criteria for diagnosis of acute myocardial infarction: Is it time for a change?" *Europace*, vol. 14, no. 1, pp. 1–9, 2012. [Online]. Available: <https://academic.oup.com/europace/article/14/1/1/493704>.
- [20] G. Hindricks et al., "2021 ESC Guidelines for the diagnosis and treatment of atrial fibrillation developed in collaboration with the European Association for Cardio-Thoracic Surgery (EACTS)," *European Heart Journal*, vol. 42, no. 5, pp. 373–498, 2021. [Online]. Available: <https://academic.oup.com/europace/article/23/5/850/6159470>.
- [21] A. L. Goldberger, "Electrocardiography," *European Heart Journal*, vol. 27, no. 10, pp. 1162–1169, 2006. [Online]. Available: <https://academic.oup.com/ehj/article/27/10/1162/2844449>.

- [22] E. M. Antman et al., "Management of acute ST-elevation myocardial infarction: A report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines," *The New England Journal of Medicine*, vol. 351, no. 24, pp. 2535–2547, 2004. [Online]. Available: <https://www.nejm.org/doi/full/10.1056/NEJMra040830>.
- [23] T. M. Mitchell, *Machine Learning*, New York: McGraw-Hill, 1997. [Online]. Available: <https://dl.acm.org/doi/book/10.5555/541177>.
- [24] C. M. Bishop, *Pattern Recognition and Machine Learning*, New York: Springer, 2006. [Online]. Available: <https://www.springer.com/gp/book/9780387310732>.
- [25] A. Y. Hannun et al., "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature Medicine*, vol. 25, no. 1, pp. 65–69, 2019. [Online]. Available: <https://www.nature.com/articles/s41591-018-0268-3>.
- [26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Cambridge: MIT Press, 2016. [Online]. Available: <https://www.deeplearningbook.org/>.
- [27] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. [Online]. Available: <https://www.nature.com/articles/nature14539>.
- [28] P. Rajpurkar et al., "Cardiologist-level arrhythmia detection with convolutional neural networks," *Nature Medicine*, vol. 25, no. 1, pp. 65–69, 2019. [Online]. Available: <https://www.nature.com/articles/s41591-018-0268-3>.
- [29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986. [Online]. Available: <https://www.nature.com/articles/323533a0>.
- [30] U. R. Acharya, H. Fujita, S. L. Oh, Y. Hagiwara, J. E. Tan, and M. Adam, "Application of deep convolutional neural networks for automated detection of myocardial infarction using ECG signals," *Scientific Reports*, vol. 7, no. 1, p. 41338, 2017. [Online]. Available: <https://www.nature.com/articles/srep41338>.
- [31] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems," in Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 2016, pp. 7–10. [Online]. Available: <https://dl.acm.org/doi/10.1145/2988450.2988454>.
- [32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learning Representations (ICLR), San Diego, CA, USA, 2015. [Online]. Available: <https://www.robots.ox.ac.uk/~vgg/publications/2015/Simonyan15/>.

- [33] Y. Zheng, S. Liu, and W. Chen, "ECG classification using convolutional neural networks," *IEEE Access*, vol. 8, pp. 101200–101209, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/8959500>.
- [34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: <https://www.jmlr.org/papers/volume3/gers02a/gers02a.pdf>.
- [35] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017. [Online]. Available: <https://ieeexplore.ieee.org/document/7508408>.
- [36] M. R. Cowie, M. A. Bax, S. Bruining, A. Cleland, M. Koehler, S. Malik, and A. E. Wood, "e-Health: a position statement of the European Society of Cardiology," *European Heart Journal*, vol. 38, no. 1, pp. 63–69, 2017. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5541086/>.
- [37] S. R. Steinhubl, E. D. Muse, and E. J. Topol, "The emerging field of mobile health," *JAMA*, vol. 313, no. 14, pp. 1416–1417, 2015. [Online]. Available: <https://jamanetwork.com/journals/jama/fullarticle/2274444>.
- [38] D. Chauhan and L. Vig, "Anomaly detection in ECG time signals via deep long short-term memory networks," in *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2015, pp. 1–7. [Online]. Available: <https://ieeexplore.ieee.org/document/7047490>.
- [39] L. Van den Bulcke, P. Willems, and P. Vandervoort, "Wearable devices in chronic disease management: Promises and pitfalls," *Sensors*, vol. 19, no. 7, p. 1569, 2019. [Online]. Available: <https://www.mdpi.com/1424-8220/19/7/1569>.
- [40] V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017. [Online]. Available: <https://ieeexplore.ieee.org/document/7920332>.
- [41] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," *arXiv preprint arXiv:1611.06440*, 2016. [Online]. Available: <https://arxiv.org/abs/1611.06440>.
- [42] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. P. Steiner, "Quantization and training of neural networks for efficient integer-arithmatic-only inference," *arXiv preprint arXiv:1712.05877*, 2018. [Online]. Available: <https://arxiv.org/abs/1712.05877>.
- [43] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015. [Online]. Available: <https://arxiv.org/abs/1503.02531>.

- [44] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2015, pp. 1737–1746. [Online]. Available: <https://arxiv.org/abs/1502.02551>.
- [45] M. Nagel, M. van Baalen, T. Blankevoort, and M. Welling, "Data-free quantization through weight equalization and bias correction," *arXiv preprint arXiv:1906.04721*, 2019. [Online]. Available: <https://arxiv.org/abs/1906.04721>.
- [46] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS)*, 2016, pp. 4107–4115. [Online]. Available: <https://arxiv.org/abs/1603.05279>.
- [47] Z. Shen, M. Zhang, J. Li, and Y. Zhang, "A reduced-precision implementation of deep neural network using dynamic quantization," in *Proceedings of the IEEE International Conference on Big Data (Big Data)*, 2018, pp. 330–339. [Online]. Available: <https://arxiv.org/abs/1810.08863>.
- [48] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," *arXiv preprint arXiv:1806.08342*, 2018. [Online]. Available: <https://arxiv.org/abs/1806.08342>.
- [49] D. Lin, S. Talathi, and S. Annapureddy, "Fixed point quantization of deep convolutional networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2016, pp. 2849–2858. [Online]. Available: <https://arxiv.org/abs/1611.03112>.
- [50] E. A. Perez Alday, A. Gu, A. Shah, C. Robichaux, A. K. I. Wong, C. Liu, F. Liu, A. B. Rad, A. Elola, S. Seyedi, Q. Li, A. Sharma, G. D. Clifford, and M. A. Reyna, "Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020," *Computing in Cardiology*, 2020. [Online]. Available: <https://physionet.org/content/challenge-2020/1.0.2/papers/#files-panel>
- [51] F. Liu, C. Liu, L. Zhao, X. Zhang, X. Wu, X. Xu, Y. Liu, C. Ma, S. Wei, Z. He, et al., "An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection", *Journal of Medical Imaging and Health Informatics*, vol. 8, no. 7, pp. 1368–1373, 2018.
- [52] V. Tihonenko, A. Khaustov, S. Ivanov, A. Rivin, and E. Yakushenko, "St Petersburg INCART 12-lead arrhythmia database", *PhysioBank, PhysioToolkit, and PhysioNet*, 2008
- [53] P. Wagner, N. Strodthoff, R.-D. Boussejot, D. Kreiseler, F. I. Lunze, W. Samek, and T. Schaeffter, "PTB-XL, a large publicly available electrocardiography dataset", *Scientific Data*, vol. 7, no. 1, pp. 1–15, 2020.

- [54] A. Natarajan, Y. Chang, S. Mariani, A. Rahman, G. Boverman, S. Vij, and J. Rubin, "A wide and deep transformer neural network for 12-lead ECG classification," in *PhysioNet/Computing in Cardiology Challenge 2020*, Cambridge, MA, USA, 2020.
- [55] Z. Zhao, H. Fang, S. D. Relton, R. Yan, Y. Liu, Z. Li, J. Qin, and D. C. Wong, "Adaptive lead weighted ResNet trained with different duration signals for classifying 12-lead ECGs," in *PhysioNet/Computing in Cardiology Challenge 2020*, Manchester, UK, 2020.
- [56] Z. Zhu, H. Wang, T. Zhao, Y. Guo, Z. Xu, Z. Liu, S. Liu, X. Lan, and M. Feng, "Classification of cardiac abnormalities from ECG signals using SE-ResNet," in *PhysioNet/Computing in Cardiology Challenge 2020*, Singapore, 2020.
- [57] M. P. Oppelt, M. Riehl, F. P. Kemeth, and J. Steffan, "Combining scatter transform and deep neural networks for multilabel electrocardiogram signal classification," in *PhysioNet/Computing in Cardiology Challenge 2020*, 2020.
- [58] H. Hasani, A. Bitarafan, and M. Soleymani Baghshah, "Classification of 12-lead ECG signals with adversarial multi-source domain generalization," in *PhysioNet/Computing in Cardiology Challenge 2020*, Tehran, Iran, 2020.
- [59] Y. Neuvo, Dong Cheng-Yu and S. Mitra, "Interpolated finite impulse response filters," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 3, pp. 563-570, June 1984

Appendix: MAX78000 and ai8x Module for Model Quantization

As part of the research for this thesis, I explored the use of the MAX78000 microcontroller and the ai8x quantization framework developed by Analog Devices. These tools are designed to facilitate the deployment of deep learning models in resource-constrained environments, such as wearables or other edge AI devices, which are highly relevant for ECG monitoring applications.

MAX78000 Overview

The MAX78000 is a specialized microcontroller designed for ultra-low-power neural network inference, particularly targeting applications at the edge of the Internet of Things (IoT). It features an integrated Arm Cortex-M4 processor alongside a Convolutional Neural Network (CNN) accelerator. The CNN accelerator allows the MAX78000 to execute AI tasks, such as keyword spotting, image recognition, or signal processing, with minimal power consumption. The MAX78000's CNN engine is optimized for 8-bit quantized models, providing significant

efficiency improvements compared to traditional floating-point operations. This makes it ideal for devices like wearables, which need to operate on battery power for extended periods. Its ability to perform real-time inference with minimal latency is another benefit for edge AI applications like continuous ECG monitoring

ai8x Training and Quantization Module

To facilitate model deployment on the MAX78000, Analog Devices developed the ai8x framework, a PyTorch-based toolkit that enables model training, fine-tuning, and quantization specifically for the MAX78xxx series of microcontrollers. The ai8x module streamlines the process of converting a trained deep learning model into a form that can be run efficiently on the MAX78000's CNN accelerator. It achieves this by reducing the precision of weights and activations from 32-bit floating-point to 8-bit integers, a process known as quantization.

However, despite the advantages of the MAX78000 and the ai8x module, several limitations impacted my work. Specifically:

1. **Limited Multi-label Classification Support:** The ai8x module is built on an older version of PyTorch, which does not support multi-label classification tasks. This posed a significant challenge, as my models are designed to classify multiple cardiac conditions simultaneously. The ai8x module only supports multi-class classification, where a single label is predicted, limiting its applicability for my use case.
2. **Conv2D Layer Quantization:** While the ai8x module is effective for many applications, it currently lacks support for quantizing Conv2D layers. Since Conv2D layers are integral to the architectures I developed for ECG signal processing, this limitation prevented the full integration of my models into the ai8x quantization pipeline

Architectures Explored with ai8x

Despite these limitations, I explored several convolutional architectures, which are worth noting. Some of these models included:

Model Architecture History

1. CNN Model 1

Hyperparameters:

Batch size: 128

Shuffle: Yes

Number of Epochs: 1000

Early Stopping: Yes

Early Stopping Patience: 10

Normalization: Yes

Normalization Method: Z – Normalization

Model Architecture:

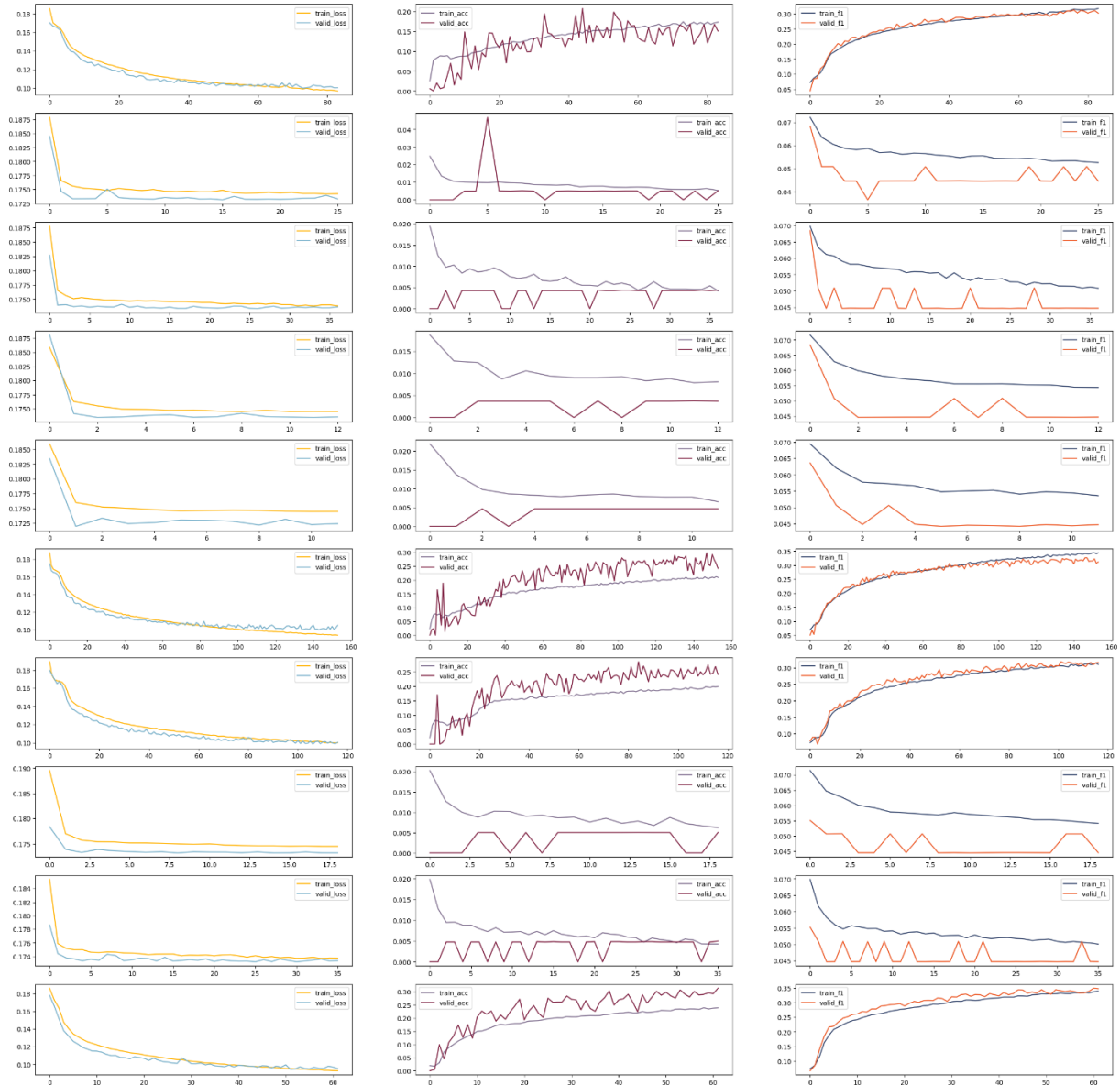
```
class ECGConvNet(nn.Module):
    def __init__(self):
        super(ECGConvNet, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, kernel_size=(1, 11), padding=(0, 5), stride=(1, 5))
        self.relu1 = nn.ReLU()
        self.maxpool1 = nn.MaxPool2d(kernel_size=(1, 8))
        self.dropout1 = nn.Dropout(p=0.2)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=(1, 11), padding=(0, 5))
        self.relu2 = nn.ReLU()
        self.maxpool2 = nn.MaxPool2d(kernel_size=(1, 4))
        self.dropout2 = nn.Dropout(p=0.2)
        self.conv3 = nn.Conv2d(64, 96, kernel_size=(1, 11), padding=(0, 5))
        self.relu3 = nn.ReLU()
        self.maxpool3 = nn.MaxPool2d(kernel_size=(1, 4))
        self.dropout3 = nn.Dropout(p=0.2)
        self.flatten = nn.Flatten()
        self.fc1 = nn.Linear(12672, 64)
        self.relu3 = nn.ReLU()
        self.dropout4 = nn.Dropout(p=0.2)
        self.fc2 = nn.Linear(64, 27)

    def forward(self, x):
        x = self.conv1(x)
```

```
x = self.relu1(x)
x = self.maxpool1(x)
x = self.dropout1(x)
x = self.conv2(x)
x = self.relu2(x)
x = self.maxpool2(x)
x = self.dropout2(x)
x = self.conv3(x)
x = self.relu3(x)
x = self.maxpool3(x)
x = self.dropout3(x)
x = self.flatten(x)
x = self.fc1(x)
x = self.relu3(x)
x = self.dropout4(x)
x = self.fc2(x)
return x
```

Number of Trainable parameters: 903,483

Evaluation Results:



2. CNN Model 2

Hyperparameters:

Batch size: 128

Shuffle: Yes

Number of Epochs: 1000

Early Stopping: Yes

Early Stopping Patience: 10

Normalization: Yes

Normalization Method: Z – Normalization

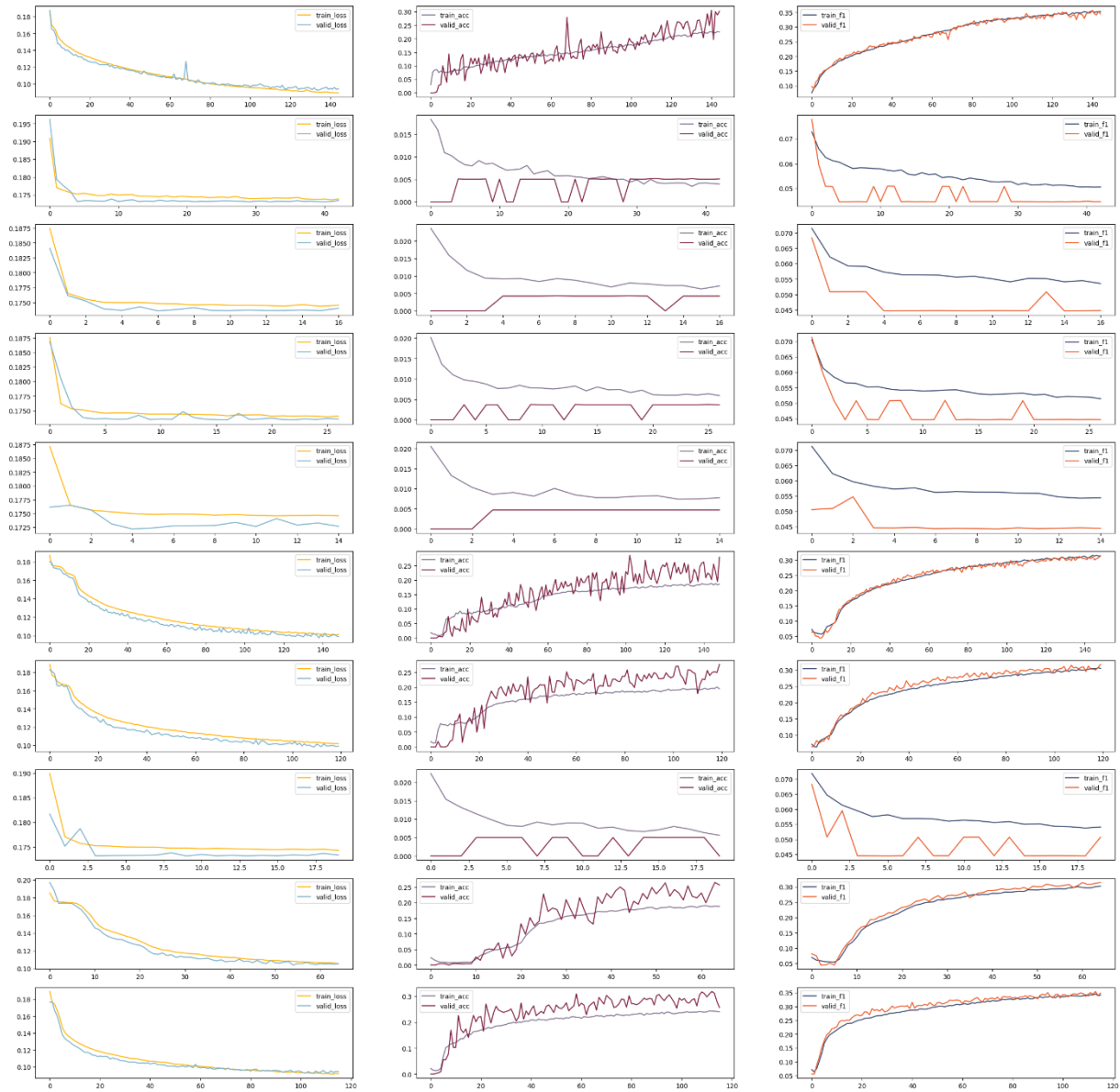
Architecture:

```
class ECGConvNet(nn.Module):
    def __init__(self):
        super(ECGConvNet2, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, kernel_size=(1, 11), padding=(0, 5), stride=(1, 5))
        self.relu1 = nn.ReLU()
        self.maxpool1 = nn.MaxPool2d(kernel_size=(1, 8))
        self.dropout1 = nn.Dropout(p=0.2)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=(1, 11), padding=(0, 5), stride=(1, 2))
        self.relu2 = nn.ReLU()
        self.maxpool2 = nn.MaxPool2d(kernel_size=(1, 4))
        self.dropout2 = nn.Dropout(p=0.2)
        self.conv3 = nn.Conv2d(64, 96, kernel_size=(1, 11), padding=(0, 5), stride=(1, 2))
        self.relu3 = nn.ReLU()
        self.maxpool3 = nn.MaxPool2d(kernel_size=(1, 4))
        self.dropout3 = nn.Dropout(p=0.2)
        self.flatten = nn.Flatten()
        self.fc1 = nn.Linear(3456, 64)
        self.relu3 = nn.ReLU()
        self.dropout4 = nn.Dropout(p=0.2)
        self.fc2 = nn.Linear(64, 27)

    def forward(self, x):
        x = self.conv1(x)
        x = self.relu1(x)
        x = self.maxpool1(x)
        x = self.dropout1(x)
        x = self.conv2(x)
        x = self.relu2(x)
        x = self.maxpool2(x)
        x = self.dropout2(x)
        x = self.conv3(x)
        x = self.relu3(x)
        x = self.maxpool3(x)
        x = self.dropout3(x)
        x = self.flatten(x)
        x = self.fc1(x)
        x = self.relu3(x)
        x = self.dropout4(x)
        x = self.fc2(x)
        return x
```

Number of Trainable parameters: 313,659

Evaluation Results:



3. CNN Model 3

Hyperparameters:

Batch size: 128

Shuffle: Yes

Number of Epochs: 1000

Early Stopping: Yes

Early Stopping Patience: 10

Lightweight Model Development for Arrhythmia Detection in Wearable Devices Using Deep Learning

Normalization: Yes

Normalization Method: Min Max Scaling

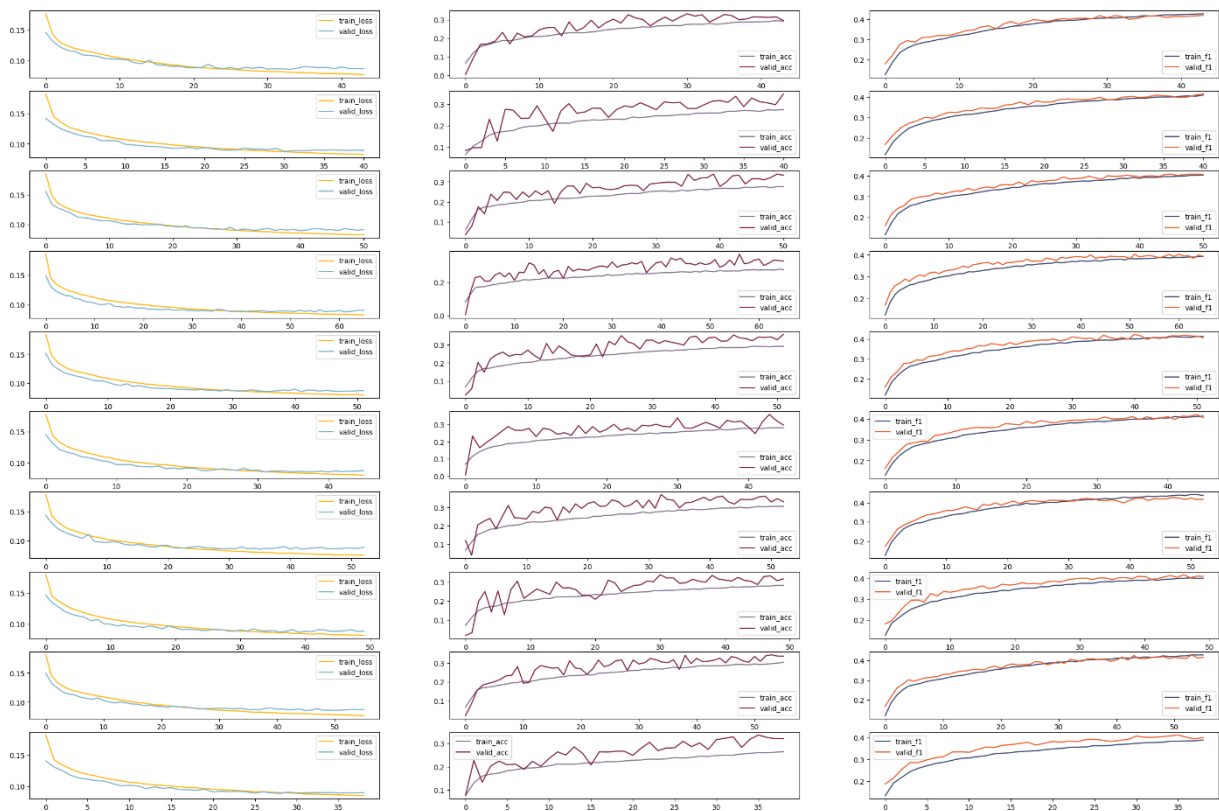
Architecture:

Same as 2.

Number of Trainable parameters:

Same as 2.

Evaluation Results:



4. CNN Model 4

Hyperparameters:

Batch size: 128

Shuffle: Yes

Number of Epochs: 1000

Early Stopping: Yes

Early Stopping Patience: 10

Normalization: Yes

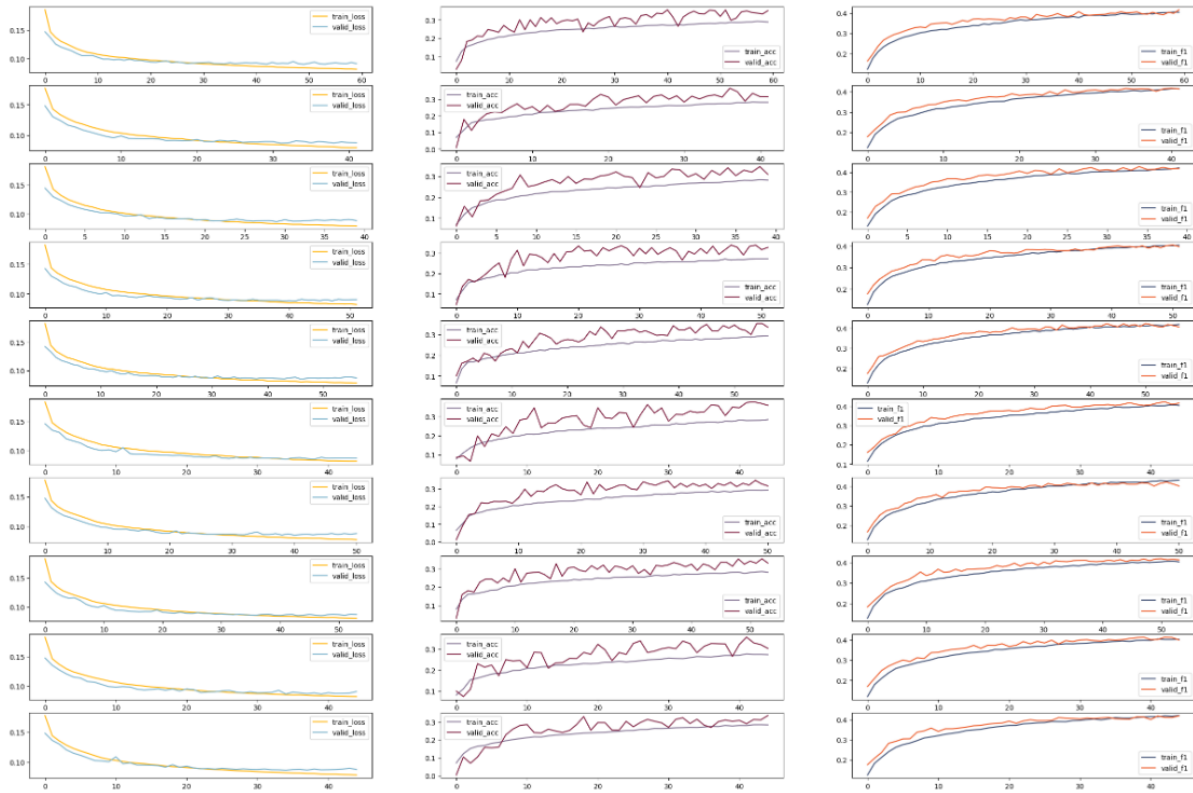
Normalization Method: Min Max Scaling

```
class ECGConvNet(nn.Module):
    def __init__(self):
        super(ECGConvNet3, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, kernel_size=(1, 9), padding=(0, 5), stride=(1, 5))
        self.relu1 = nn.ReLU()
        self.maxpool1 = nn.MaxPool2d(kernel_size=(1, 8))
        self.dropout1 = nn.Dropout(p=0.2)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=(1, 11), padding=(0, 5), stride=(1, 2))
        self.relu2 = nn.ReLU()
        self.maxpool2 = nn.MaxPool2d(kernel_size=(1, 4))
        self.dropout2 = nn.Dropout(p=0.2)
        self.conv3 = nn.Conv2d(64, 96, kernel_size=(1, 11), padding=(0, 5), stride=(1, 2))
        self.relu3 = nn.ReLU()
        self.maxpool3 = nn.MaxPool2d(kernel_size=(1, 4))
        self.dropout3 = nn.Dropout(p=0.2)
        self.flatten = nn.Flatten()
        self.fc1 = nn.Linear(3456, 64)
        self.relu3 = nn.ReLU()
        self.dropout4 = nn.Dropout(p=0.2)
        self.fc2 = nn.Linear(64, 27)

    def forward(self, x):
        x = self.conv1(x)
        x = self.relu1(x)
        x = self.maxpool1(x)
        x = self.dropout1(x)
        x = self.conv2(x)
        x = self.relu2(x)
        x = self.maxpool2(x)
        x = self.dropout2(x)
        x = self.conv3(x)
        x = self.relu3(x)
        x = self.maxpool3(x)
        x = self.dropout3(x)
        x = self.flatten(x)
        x = self.fc1(x)
        x = self.relu3(x)
        x = self.dropout4(x)
        x = self.fc2(x)
        return x
```

Number of Trainable parameters: 313,595

Evaluation Results:



5. CNN Model 5

Hyperparameters:

Batch size: 128

Shuffle: Yes

Number of Epochs: 1000

Early Stopping: Yes

Early Stopping Patience: 10

Normalization: Yes

Normalization Method: Min Max Scaling

Architecture:

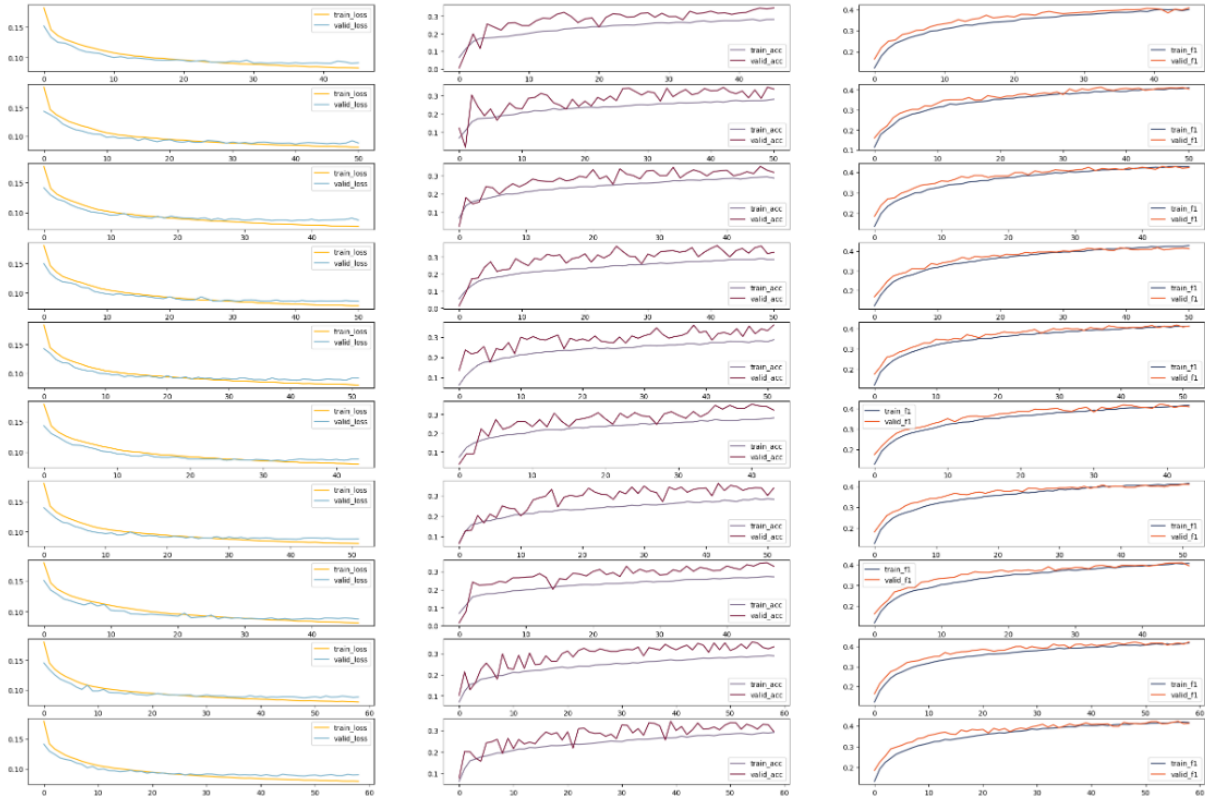
```
class ECGConvNet(nn.Module):
    def __init__(self):
        super(ECGConvNet4, self).__init__()
```

```
self.conv1 = nn.Conv2d(1, 32, kernel_size=(1, 7), padding=(0, 5), stride=(1, 5))
self.relu1 = nn.ReLU()
self.maxpool1 = nn.MaxPool2d(kernel_size=(1, 8))
self.dropout1 = nn.Dropout(p=0.2)
self.conv2 = nn.Conv2d(32, 64, kernel_size=(1, 11), padding=(0, 5), stride=(1, 2))
self.relu2 = nn.ReLU()
self.maxpool2 = nn.MaxPool2d(kernel_size=(1, 4))
self.dropout2 = nn.Dropout(p=0.2)
self.conv3 = nn.Conv2d(64, 96, kernel_size=(1, 11), padding=(0, 5), stride=(1, 2))
self.relu3 = nn.ReLU()
self.maxpool3 = nn.MaxPool2d(kernel_size=(1, 4))
self.dropout3 = nn.Dropout(p=0.2)
self.flatten = nn.Flatten()
self.fc1 = nn.Linear(3456, 64)
self.relu3 = nn.ReLU()
self.dropout4 = nn.Dropout(p=0.2)
self.fc2 = nn.Linear(64, 27)

def forward(self, x):
    x = self.conv1(x)
    x = self.relu1(x)
    x = self.maxpool1(x)
    x = self.dropout1(x)
    x = self.conv2(x)
    x = self.relu2(x)
    x = self.maxpool2(x)
    x = self.dropout2(x)
    x = self.conv3(x)
    x = self.relu3(x)
    x = self.maxpool3(x)
    x = self.dropout3(x)
    x = self.flatten(x)
    x = self.fc1(x)
    x = self.relu3(x)
    x = self.dropout4(x)
    x = self.fc2(x)
    return x
```

Number of Trainable parameters: 313,531

Evaluation Results:



6. CNN Model 6

Hyperparameters:

Batch size: 128

Shuffle: Yes

Number of Epochs: 1000

Early Stopping: Yes

Early Stopping Patience: 30

Normalization: Yes

Normalization Method: Min Max Scaling

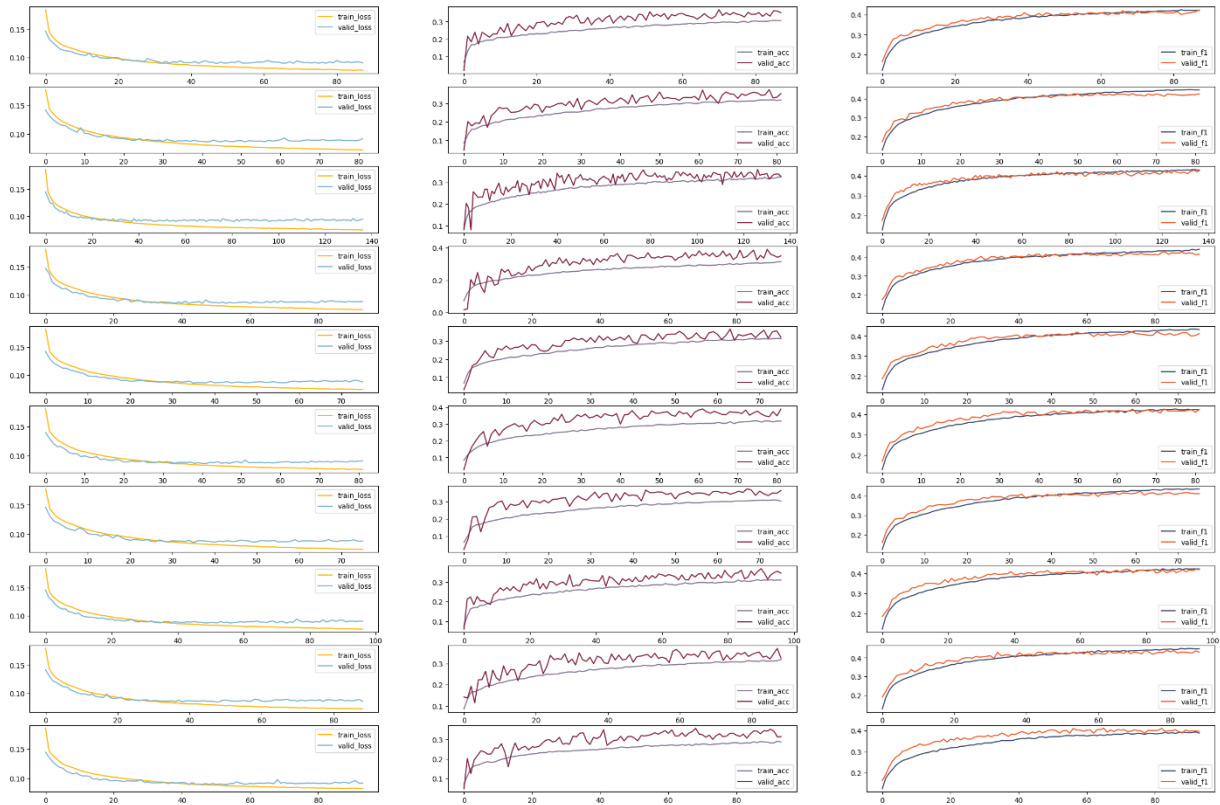
Architecture:

Same as 5.

Number of Trainable parameters:

Same as 5.

Evaluation Results:



7. CNN Model 7

Hyperparameters:

Batch size: 128

Shuffle: Yes

Number of Epochs: 1000

Early Stopping: Yes

Early Stopping Patience: 50

Normalization: Yes

Normalization Method: Min Max Scaling

Architecture:

```
class ECGConvNet(nn.Module):
    def __init__(self):
        super(ECGConvNet5, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, kernel_size=(1,7), padding=(0, 5), stride=(1, 5))
        self.relu1 = nn.ReLU()
        self.maxpool1 = nn.MaxPool2d(kernel_size=(1, 8))
```

```
self.dropout1 = nn.Dropout(p=0.2)
self.conv2 = nn.Conv2d(32, 32, kernel_size=(1, 11), padding=(0, 5), stride=(1, 2))
self.relu2 = nn.ReLU()
self.maxpool2 = nn.MaxPool2d(kernel_size=(1, 4))
self.dropout2 = nn.Dropout(p=0.2)
self.conv3 = nn.Conv2d(32, 64, kernel_size=(1, 11), padding=(0, 5), stride=(1, 2))
self.relu3 = nn.ReLU()
self.maxpool3 = nn.MaxPool2d(kernel_size=(1, 4))
self.dropout3 = nn.Dropout(p=0.2)
self.flatten = nn.Flatten()
self.fc1 = nn.Linear(2304, 64)
self.relu3 = nn.ReLU()
self.dropout4 = nn.Dropout(p=0.2)
self.fc2 = nn.Linear(64, 27)

def forward(self, x):
    x = self.conv1(x)
    x = self.relu1(x)
    x = self.maxpool1(x)
    x = self.dropout1(x)
    x = self.conv2(x)
    x = self.relu2(x)
    x = self.maxpool2(x)
    x = self.dropout2(x)
    x = self.conv3(x)
    x = self.relu3(x)
    x = self.maxpool3(x)
    x = self.dropout3(x)
    x = self.flatten(x)
    x = self.fc1(x)
    x = self.relu3(x)
    x = self.dropout4(x)
    x = self.fc2(x)
    return x
```

Number of Trainable parameters: 183,419

Evaluation Results:

