NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF PHYSICS

LEARNING FROM BIOSIGNALS

A DEEP NEURAL APPROACH TO ADVANCED SIGNAL PROCESSING

MARIANA DE AVELINO GERALDO DIAS

Master in Biomedical Engineering

DOCTORATE IN BIOMEDICAL ENGINEERING

NOVA University Lisbon

May, 2025

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
PHYSICS

LEARNING FROM BIOSIGNALS

A DEEP NEURAL APPROACH TO ADVANCED SIGNAL PROCESSING

MARIANA DE AVELINO GERALDO DIAS

Master in Biomedical Engineering

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

Examination Committee

Chair: Maria de Fátima Guerreiro da Silva Campos Raposo
Full Professor, NOVA University Lisbon

Rapporteurs: Michael Wand
Senior Researcher, Scuola Universitaria Professionale della Svizzera Italiana

Inês Nunes de Sousa Soares
Senior Researcher, Fraunhofer Portugal AICOS

Members: Maria de Fátima Guerreiro da Silva Campos Raposo
Full Professor, NOVA University Lisbon

Ricardo Nuno Pereira Verga e Afonso Vigário
Associate Professor, NOVA University Lisbon

DOCTORATE IN BIOMEDICAL ENGINEERING

NOVA University Lisbon
May, 2025

Learning from Biosignals
a Deep Neural Approach to Advanced Signal Processing

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.1.27) [114].

Acknowledgements

The thesis you are about to read (or perhaps just take a glimpse at) goes by my name, but is far from being mine alone. Throughout the past four years, I was lucky to cross paths with many people who, knowingly or not, share its ownership with me.

First, I owe this work to Professor Hugo Gamboa. He was not only the one who consistently challenged me to go further, but also the one who made me believe I could. For your constant guidance, your knowledge, and your time - thank you.

To everyone from my Libphys-Biosignals family - thank you for being such a fundamental part of this journey. To Luís, a great mentor who taught me a lot and whom I knew I could always count on. To Phillip, my partner in the craziness of pursuing a PhD, always so generous to everyone around him, and the best critic who never misses a detail. To the other partners who joined later, Inês and Dania, and to those who are no longer in the lab but who positively impacted the first years of my PhD with their experience and kindness — Cátia, João, Daniel, Federico, and Maria Eduarda. To Márcia and Beatriz, the master's students I was lucky enough to accompany closely. I was fortunate to be surrounded by an incredible team whose strong sense of community made each workday feel lighter.

I would like to thank Fundação para a Ciência e Tecnologia for funding this work and MIT Portugal for having selected this project for funding and for the support and opportunities given. To everyone involved in the Operator Project, an MPP Flagship Project through which I had the opportunity to collaborate with Autoeuropa, Fraunhofer, Universidade do Minho, MIT, among other institutions.

I also owe a big thank you to Fulbright Portugal, to Praneeth Namburi and Brian Anthony from MIT.Nano, and to everyone from the Immersion Lab. The four months I spent at MIT were among the most incredible experiences my path in science has offered me so far.

A huge thank you to my university, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, and to the Physics Department, where I spent a big part of my life over the last 4 years. It was also where, alongside my PhD, I discovered something I never expected to have so much fun doing: teaching. Thank you to the Professors who introduced me to this new craft, and to all the students who came across my classes.

To all my friends, without whom i wouldn't have survived this PhD. Specially to Carol, Maria, Bia, Mafalda, Nana, Cris, Ricardo, Joana, Rodrigo, Vale, Alex, Guedes, Grilo, Afonso, Leonor, Carvalho, and Bruno - in the most stressful times, what kept me going was knowing our next meetup wasn't far off.

Last, but most importantly, to my family - mom, dad, Afonso, and Camila - obrigada!

Abstract

When it comes to biosignal processing, a practitioner must undergo a series of steps to prepare the data for analysis. These foundational tasks include assessing data integrity, executing cleansing protocols, standardizing measurements through normalization and unit conversion, and isolating features for the intended analysis. While there are pipelines that guide these procedures, these need to be customized in order to not only adjust to the different characteristics of each dataset but also to the information that is to be extracted. The customization relies on the practitioner's know-how, underscoring the importance of expertise in transforming raw biosignals into meaningful conclusions.

Meanwhile, Deep Learning (DL) frameworks for time-series have consistently demonstrated great potential in performing tasks such as classification and processing of biosignals. For that reason, the present work focuses on the application of DL methodologies for biosignals. We start by addressing specific tasks such as disease classification and noise removal in ECG signals. Various strategies are explored, including alternative data representations and architectural paradigms, with a focus on efficiency through reduced model complexity. Additionally, a custom data collection was conducted in industrial environments, and the resulting dataset was used to evaluate the generalization capacity of the noise removal model across different acquisition conditions.

Given that a robust DL model needs to learn multiple signal features to, for instance, distinguish between normal and pathological states, leveraging this acquired knowledge across various processing tasks could potentially revolutionize our approach in biosignal analysis, making the case for a more integrated and versatile deployment of DL techniques. In this context, we developed `NeuralLib`, a framework that provides functionalities to train, reuse and access networks designed to perform biosignal processing functions, built on three key principles: modularity, efficiency, and generalization. These networks learn the fundamental characteristics of the signals, and by fine-tuning them for new tasks, it becomes possible to transfer prior knowledge, enabling not only faster convergence but also reducing the need for large annotated datasets.

Keywords: Biosignals, Deep Learning, ECG, Modularity, Efficiency, Generalization

Resumo

No processamento de biossinais, é necessário realizar uma série de etapas para preparar os dados para análise. Estas etapas fundamentais incluem a avaliação da integridade dos dados, a aplicação de protocolos de limpeza, a normalização e conversão dos sinais, bem como a extração de características relevantes para a análise pretendida. Embora haja procedimentos relativamente padronizados, estes necessitam frequentemente de ser adaptados, tanto às particularidades de cada conjunto de dados, como ao tipo de informação que se pretende extrair. A análise de biossinais requer, por isso, alguma experiência e conhecimento acerca destes, para que se consiga transformá-los em informação relevante.

Paralelamente, modelos de Aprendizagem Profunda (AP) para séries temporais têm demonstrado elevado potencial na resolução de tarefas como classificação e processamento de biossinais. O presente trabalho é, por isso, centrado na aplicação de metodologias de AP para biossinais. São desenvolvidas, inicialmente, tarefas específicas, como a classificação de doenças e a remoção de ruído a partir de sinais de ECG, sendo diversas estratégias exploradas - incluindo diferentes formatos de representação dos dados e diferentes arquiteturas - como ênfase na eficiência através da redução da complexidade dos modelos. Adicionalmente, foi realizada uma recolha de dados em contexto industrial, de forma a avaliar a capacidade de generalização do modelo de remoção de ruído em diferentes contextos de aquisição.

Considerando que um modelo de AP robusto precisa de aprender múltiplas características do sinal para, por exemplo, distinguir entre estados normais e patológicos, a reutilização deste conhecimento adquirido em diferentes tarefas de processamento poderá transformar a abordagem atual de análise de biossinais, sustentando a importância de uma aplicação mais integrada e versátil das técnicas de AP. Neste contexto, desenvolvemos `NeuralLib`, um *software* que inclui as funcionalidades para treinar, reutilizar e aceder a redes neuronais para executar funções de processamento de biossinais, assente em três princípios fundamentais: modularidade, eficiência e generalização. Estas redes aprendem as características fundamentais dos sinais e, através de técnicas de retreino para novas tarefas, torna-se possível transferir conhecimento previamente adquirido, permitindo não só uma convergência mais rápida, como também a redução da necessidade de grandes

volumes de dados anotados.

Palavras-chave: Biossinais, Aprendizagem Profunda, ECG, Modularidade, Eficiência, Generalização

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence *(pp. 3, 4, 12, 20, 21, 33–36, 42–44, 92)*
ANN Artificial Neural Network *(pp. 21, 22)*

BiGRU Bidirectional Gated Recurrent Unit *(pp. 53, 59, 75, 77, 82, 84)*
BiLSTM Bidirectional Long Short Term Memory *(pp. 40, 53, 59)*
BLA Blood Lactate Accumulation *(pp. 16, 17)*
BP Blood Pressure *(pp. 16, 17)*
BW Baseline Wander *(pp. xi, 67–69, 71, 72, 78, 79, 82–84, 86–88)*

CD Conduction Disturbance *(p. 47)*
CNN Convolutional Neural Network *(pp. xi, xiii, 3, 25, 29, 31, 39, 40, 48–51, 53–55, 59, 60, 63, 65, 69, 70, 85, 86, 104)*
CVD cardiovascular disease *(pp. 4, 8, 14–16, 18, 45–48, 51, 55, 57, 62, 63, 67, 103, 104)*
CVL cardiovascular load *(pp. xiii, 14–16, 18)*
CVS Cardiovascular Strain *(pp. 15, 16)*

DA Domain Adaptation *(pp. 32, 94, 95)*
DG Domain Generalization *(pp. 32, 94, 95)*
DL Deep Learning *(pp. 2–5, 19–21, 23, 24, 29, 31–38, 40–48, 50, 61–63, 65, 67, 69, 73, 84, 85, 88–96, 98, 99, 101–107, 109, 110)*

ECG Electrocardiogram *(pp. x–xii, 1, 2, 4, 5, 7, 8, 10–14, 16, 17, 29, 40–43, 45–58, 60–77, 81, 82, 84–86, 88, 103, 104, 106–108)*
EEG Electroencephalogram *(pp. x, 1, 7–11, 13, 29, 42, 43)*
EM Electrode Motion *(pp. xi, 67–69, 71, 72, 78, 79, 82–84, 86–88)*
EMG Electromyogram *(pp. x, 1, 7, 8, 10, 11, 13, 14, 29, 40–43, 68)*

FFNN Feed Forward Neural Network *(pp. x, 21, 22, 24, 26, 69, 84)*

GAF Gramian Angular Field *(pp. xi, 48–51, 54, 55, 62, 63)*
GRU Gated Recurrent Unit *(pp. x, xi, xiii, 27, 28, 40, 48, 51, 53, 55, 56, 59, 61–63, 65, 69, 72, 73, 77, 79, 80, 84, 86, 99, 101, 104)*

HR Heart Rate *(pp. 8, 11–18)*
HRR Heart Rate Reserve *(p. 16)*
HRV Heart Rate Variability *(pp. 8, 10, 11, 15–17, 42)*
HYP Hypertrophy *(pp. 47, 52, 64, 66)*

IMU Inertial Measurement Unit *(pp. 9, 14)*

LSTM Long Short Term Memory *(pp. x, 26–28, 40, 48, 50, 51, 53, 59, 63, 69, 70, 85, 104)*

MA Muscle Activation *(pp. xi, 67–69, 71, 72, 78, 82–84, 87, 88)*
MI Myocardial Infarction *(pp. 47, 50, 52, 63, 64)*
ML Machine Learning *(pp. 3, 20–22, 31, 32, 37)*
MTF Markov Transition Field *(pp. xi, 48–51, 54, 55, 62, 63)*

NLP Natural Language Processing *(pp. 30, 42)*

PPG Photoplethysmography *(pp. 40, 43)*
PRD Percentage-Root-Mean-Square Difference *(pp. xiii, xiv, 73, 77–79)*
PSI Physiological Strain Index *(pp. 16, 17)*

RHR Relative Heart Rate *(pp. 15, 16)*
RIP Respiratory Inductance Plethysmography *(pp. x, 7, 9, 10)*
RMSE Root-Mean-Square Error *(pp. xiii, xiv, 73, 77–79, 82)*
RNN Recurrent Neural Network *(pp. x, xi, xiii, 3, 25–29, 31, 40, 51, 53, 54, 59, 63, 86, 104)*
RP Recurrence Plot *(pp. xi, 48–51, 54, 55, 62, 63)*
RVO_2 Relative Oxygen consumption *(p. 16)*

SNR Signal-to-Noise Ratio *(pp. xiii, xiv, 72, 73, 77–79, 82)*
SSL Self-Supervised Learning *(pp. 22, 36, 37, 41, 43)*
STTC ST/T Changes *(pp. 47, 52, 63)*

TL Transfer Learning *(pp. xii, 3, 5, 32, 37–41, 43, 89, 91, 93–96, 100, 101, 105, 108)*

VO_2 Oxygen consumption *(pp. 16, 17)*

1

Introduction

1.1 Motivation

1.1.1 Biosignals as Windows into Human Physiology

Biosignals are timeseries generated by living systems, serving as measurable representations of physiological processes [167]. Whether it's the pulses of an Electrocardiogram (ECG) tracking heartbeats, the waves of an Electroencephalogram (EEG) capturing brain activity, or the muscle contractions detected by an Electromyogram (EMG), biosignals allow us to observe in detail complex processes of human physiology.

Biosignals have long been indispensable in healthcare, enabling the monitoring of vital signs [128], the early detection of diseases [198, 80], or the assessment of treatment efficacy [113], this way supporting the adaptation of interventions according to each individual's physiological responses. More recently, wearable technologies have brought biosignal acquisition beyond clinical walls: initially for long-term monitoring of patients outside hospital settings, but over time, these sensors have become widely adopted by a larger segment of the general population for tracking heart rate, sleep patterns, and physical activity in real time [164, 154].

The applications of biosignals have found growing relevance in other fields. Sports was among the first fields to make use of physiological monitoring, leveraging it to optimize performance, prevent injuries [107] and improve training programs [144]. In occupational settings, for instance, biosignals are increasingly used to assess physical and mental stress [21], detect fatigue, and protect worker safety [119].

Being a worker and being a patient are not mutually exclusive states and our health doesn't exist in isolation from our work environment, nor does our work performance remain unaffected by our physiological state. According to data provided by the International Labour Organization (ILO), the global average working time is approximately 8 hours per day, though it varies significantly across different regions of the world [142]. Given that a large portion of our daily lives is spent in the workplace, minimizing risks and ensuring safer conditions for all workers is crucial - wearable sensors play a key role in this process, as they enable data collections in a variety of environments. Data is essential

both for identifying correlations, establishing causal relationships, and understanding physiological responses, as well as for assessing the effectiveness of implemented measures to ensure that only those proven to be effective are maintained and replicated. However, the transition from data acquisition to information extraction is not straightforward. This is where advanced processing methods become highly relevant, and Deep Learning (DL) emerges as a powerful tool to bridge this gap.

1.1.2 Enhancing Biosignal Processing with Deep Learning

If data is the new gold, sensors are great prospecting tools, and human bodies valuable mines. Health practitioners and researchers take on the roles of miners, refiners, and jewelers, transforming raw physiological data into meaningful observations. With the democratization of access to wearable sensors, the production of this precious resource is happening at an unprecedented rate, often outpacing traditional methods of analysis. Although many wearable devices incorporate some data processing to provide feedback to users, generally it is still advisable that for clinical and research purposes there is a technician analysing the data in its raw format, since the generated information remains limited and is often unsuitable for clinical and research purposes, where high precision and validation are required [13, 48, 149]. This creates a pressing need for scalable, intelligent solutions.

Biosignals, while incredibly informative, present a host of challenges: they are inherently noisy, non-stationary, and highly variable between and within individuals. These challenges become even more pronounced when using wearables in uncontrolled environments, where subjects move freely and multiple sources of noise can interfere with the recordings. The ECG is an example of a biosignal widely used across various contexts (clinical, occupational, and sports), with numerous available devices - however, the noise contamination often limits its reliability and usefulness. There are several different challenges in biosignal processing and all these complexities make it difficult to extract useful information using conventional universal approaches, which are often limited in their ability to handle such variability and scale. Moreover, interpreting them frequently requires sophisticated techniques capable of extracting meaningful patterns from seemingly chaotic data. This is where advanced data processing methods, such as DL, come into play.

In the last decade, DL has emerged as a powerful tool for extracting relevant outputs from complex datasets. Unlike traditional signal processing techniques, which often rely on predefined assumptions and hand-crafted features, DL leverages neural networks to automatically learn patterns and representations directly from raw data. These methods not only process biosignals more effectively but also extract patterns in the data that traditional approaches possibly overlooked.

Applications of DL in biosignal analysis span a wide range of tasks, from distinguishing between physiological information and noise to detecting pathological patterns and

predicting physiological states. There are strong advantages in using DL in biosignal processing, mainly since these models are capable of handling the non-linear, noisy, and variable nature of biosignals. The existence of different types of architectures, each with different strengths, also allows us to choose those that are more adequate for the data and task to be performed. For instance, Recurrent Neural Networks (RNNs) and one-dimensional Convolutional Neural Networks (CNNs) are particularly suited for biosignal tasks, allowing for the analysis of spatial and temporal patterns: RNNs capture temporal dependencies while CNNs excel at identifying morphological features [101]. Transformers have also proved to be effective in modeling complex and long temporal patterns from biosignals, emerging as a promising paradigm in this field of research [10].

However, the adoption of DL in biosignal processing is not without challenges. These include the need for large, diverse datasets to train robust models, the risks of overfitting and weak generalization, and the persistence of data silos - isolated collections of data that are difficult to integrate with other datasets. Additionally, from an ethical perspective, even when large amounts of data are available, training large-scale models has a significant environmental impact. Both the energy consumed during training and the resources required to develop the necessary hardware are critical factors that must not be overlooked. Addressing these concerns demands great efforts to mitigate their effects and promote sustainable practices in Artificial Intelligence (AI) research.

1.1.3 Progressing Towards a Unified Framework

The concept of Machine Learning (ML) is based on the idea that, by accessing many examples, algorithms can learn how systems behave in certain scenarios. Although the differences are vast, a conceptual parallel can be drawn with how humans learn - through experience and exposure to examples. One of the key differences between the human brain and artificial neural networks is plasticity, which allows us to adapt previous knowledge to new situations without needing to see identical examples. Now, imagine if learning something new required us to erase everything we had learned before - clearly, this would make the process far more inefficient. From this intuition arises the question: why not leverage previously acquired knowledge to learn new tasks or interpret new data?

Transfer Learning (TL) provides means to accomplish this by enabling models to transfer knowledge from one domain or task to another. While not infallible, TL has been successfully applied in several cases, demonstrating that previously trained models can serve as strong foundations for new tasks. This not only improves learning efficiency but also reduces the need for extensive labeled datasets and computational resources, making AI development more sustainable. Sharing trained models with the research community also reduces redundancy and resource consumption, allowing teams to build on existing work instead of retraining similar models from scratch. This collaborative approach improves efficiency, accelerates progress, and minimizes computational and environmental costs.

Researchers and practicioners should not abstain from taking an active role in addressing AI challenges. Among these, interpretability, data biases, and environmental impact stand out as critical concerns. While the first two have received considerable attention - driven by the necessity of ensuring model fairness and trustworthiness - the third, environmental impact, has been less prominent within the AI development sphere [193]. Yet, as models grow in size and complexity, their energy consumption, carbon footprint, and reliance on extensive computational infrastructure become increasingly significant concerns that cannot be ignored. Addressing these challenges requires deliberate efforts to develop AI systems that balance high performance with sustainability, ensuring that advancements in DL do not come at the cost of ethical and environmental responsibility.

1.2 Research Questions

Biosignal processing presents a unique set of challenges due to the inherent variability, non-stationarity, and susceptibility to noise in physiological data. Traditional approaches often struggle to generalize across different datasets, highlighting the potential of data-driven methods like DL as a more effective alternative. However, while DL models have demonstrated strong performance in biosignal-related tasks, their widespread application comes with trade-offs, particularly in computational efficiency and generalization. This thesis explores how DL can be leveraged to address biosignal processing challenges while considering these trade-offs.

To frame this study, the following two Research Questions (RQ) were established:

RQ1 How can DL be effectively applied to biosignal processing tasks?

RQ2 What strategies can be implemented to improve computational efficiency in DL models while maintaining performance in biosignal applications?

1.3 Objectives

To approach the Research Questions, four main objectives were drawn:

O1: Apply DL techniques for cardiovascular disease (CVD) classification, analysing model complexity and performance. The relationship between complexity and performance is a fundamental concern for finding efficient solutions. By applying diverse DL techniques to classify CVDs from ECG data in distinct formats, this objective aims to evaluate how data format and consequent architecture choices influence performance and model complexity.

O2: Develop a DL model for ECG noise removal and assess its generalization across diverse datasets. Noise contamination is a major limiting factor in real-world biosignal applications, particularly in occupational settings where environmental conditions are

uncontrolled. This objective focuses on creating and evaluating a DL-based noise removal framework that can effectively denoise ECG signals, first in simulated noisy data and then in real-world industrial settings, to assess its ability to generalize across different acquisition conditions.

O3: Develop a systematic DL framework that supports modularity and efficiency in biosignal processing tasks. This objective seeks to develop a structured framework that allows for modular adaptation of models across different biosignal processing tasks, optimizing resource use and enabling knowledge transfer through Transfer Learning (TL) and model sharing.

O4: Ensure accessibility and scalability by making the models, training methodology, and codebase open-source for the research community. Open research is key to accelerating progress in biosignal processing and avoid unnecessary resource allocation. To facilitate widespread adoption and further development, this objective aims to make models, training methodologies, and codebases openly available to the research community.

1.4 Thesis Structure

This thesis is structured to progressively build upon the outlined research questions and objectives. It is divided into seven chapters, each addressing key aspects of the study, from theoretical foundations to experimental validation and conclusions.

Chapter 1 introduces the motivation behind the study, highlighting the challenges in biosignal processing and the role of DL in addressing them. The research questions and objectives that drove this investigation are established. Chapter 2 provides an overview of biosignals, discussing their physiological significance, acquisition methods, and challenges in processing. This chapter also explores various applications, particularly in assessing cardiovascular load in occupational settings. Chapter 3 delves into Deep Learning fundamentals, covering neural network architectures, training methodologies, optimization strategies, and challenges associated with this topic. Additionally, it presents a narrative review of deep learning applications in biosignal processing. Chapter 4 focuses on the application of deep learning techniques for cardiovascular disease classification using ECG signals. It explores the impact of model complexity on classification performance and discusses the trade-offs between accuracy and computational efficiency. Chapter 5 investigates ECG noise removal using DL models. The chapter details the development and evaluation of a denoising framework, assessing its effectiveness in both simulated and real-world industrial settings. Chapter 6 introduces the systematic DL framework developed in this work. This framework is designed to enhance modularity, facilitate model adaptation across biosignal processing tasks, and promote efficient transfer learning and

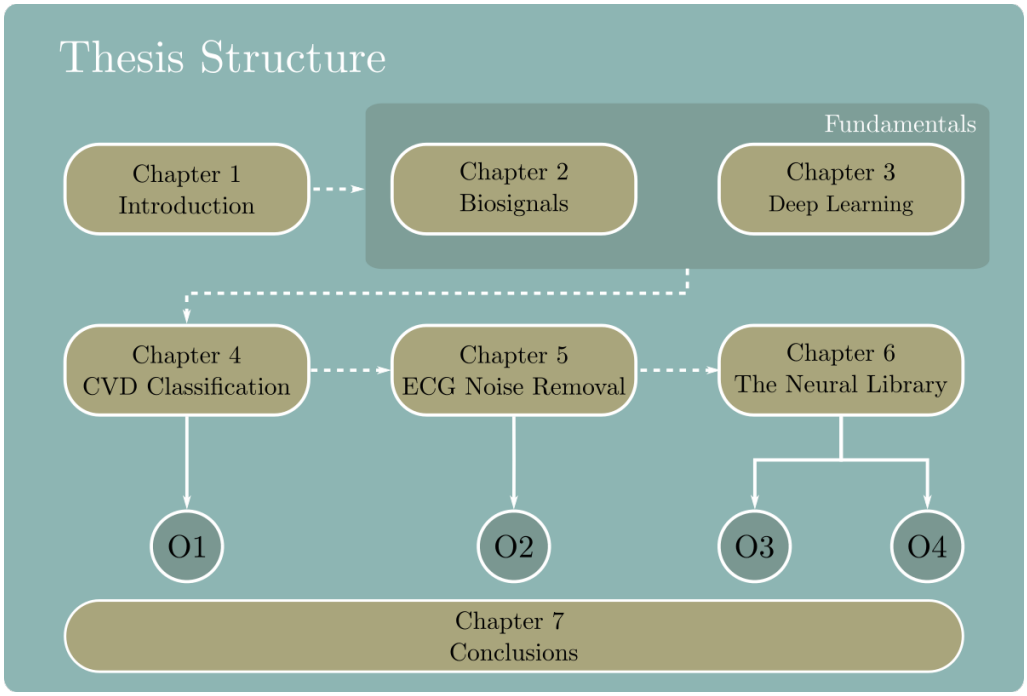


Figure 1.1: Illustration of the thesis structure.

model sharing. Finally, Chapter 7 presents the key findings of the study, summarizing contributions and discussing limitations. Future research directions are proposed.

Figure 1.1 provides a visual representation of the thesis structure, illustrating the logical flow of topics and objectives throughout the thesis.

2

Biosignals: a means to trace the body's activity

Biosignals represent the application domain of the present work. This particular kind of signals provides important information regarding the mechanisms of the human body and have been used as a means to monitor it. Several technologies have been developed to improve the level of detail captured in these signals, but its interpretation still depends on processing steps that transform them in readable metrics.

In this chapter, we will dive into the world of biosignals and understand what these are, what characterizes them and how they can be used to extract meaningful information.

2.1 What are Biosignals?

2.1.1 Definition and Examples

Biosignals are timeseries that result from measuring physiological processes. These signals are acquired through sensors, and they represent electrical, mechanical, biochemical, acoustic, thermal, or optical activities of any biological system. They are used as non-invasive forms of monitoring health, diagnosing conditions, and understanding the intricate dynamics of the human body. Examples of commonly studied biosignals include the Electrocardiogram (ECG), Electromyogram (EMG) and Electroencephalogram (EEG) (electrical signals), Respiratory Inductance Plethysmography (RIP) (a mechanical signal reflecting respiratory effort), and kinematic signals such as accelerometer and gyroscope data (which capture movement and orientation). Each type of biosignal offers unique insights into specific physiological processes, contributing to a comprehensive understanding of the body's functioning.

Any signal that reflects a living being-derived process through time can be called a biosignal. Some of the most widely studied examples are described below and illustrated in Figure 2.1.

Electrocardiogram (ECG) ECG refers both to a medical exam and to the biosignal it records. It is a record of the electrical activity of the heart and its waveform can be separated into distinct sections that can be attributed to physiological processes occurring during the cardiac cycle: (1) the P wave, generated by atrial depolarization, (2) the QRS complex, arising from ventricular depolarization, and (3) the T wave that forms due to ventricular repolarization [89]. ECG is widely used in clinical practice in the diagnosis of cardiovascular disease (CVD), since heart conditions often manifest as specific characteristic disturbances in its waveform, depending on the underlying anatomical, functional, or physiological cause of the condition. Beyond direct cardiac assessment, the signal's temporal characteristics also contain valuable physiological information. For instance, each R peak corresponds to a heartbeat, allowing the calculation of Heart Rate (HR) - the number of heartbeats per minute - and derived metrics such as Heart Rate Variability (HRV) [52]. HRV, which quantifies variations in the time intervals between consecutive R peaks (RR intervals), serves as an indicator of autonomic nervous system activity and the heart's adaptability to physiological demands [174]. In section 2.3 we further explore cardiovascular metrics, particularly in the context of cardiovascular load assessment in occupational settings. Moreover, Chapters 4 and 5 cover ECG processing challenges and more features regarding this particular biosignal are analyzed.

Electromyogram (EMG) EMG signal is the electrical manifestation of the neuromuscular activation associated with a contracting muscle [115]. It reflects muscle activation caused by neuromuscular cells. In clinical practice, EMG can help differentiate between muscle weakness caused by muscle disorders (myopathic) and that caused by nerve problems (neurogenic) [130]. EMG signals can be recorded using either invasive needle electrodes, which provide detailed information about individual motor units, or non-invasive surface electrodes (surface EMG) placed on the skin immediately above the muscle tissue. Surface EMG are widely used in research, sports science, and rehabilitation due to their ease of application. EMG convey information regarding muscle function, including the intensity of muscle contractions, recruitment of motor units, and the myoelectric manifestation of muscle fatigue [49]. Due to its inherently high frequency content, the signal is not easily interpretable by the naked eye. As such, proper signal processing techniques, often including several steps - such as rectification, filtering, and spectral analysis - are crucial to ensure reliable interpretation.

Electroencephalogram (EEG) The EEG records the electrical activity of the brain, capturing the various brain waves associated with different states such as sleep, alertness, or cognitive tasks. It is one of the standard methods to measure brain activity in various fields and serves as the primary signal source for noninvasive brain-computer interfaces (BCIs). EEG signals are recorded via electrodes placed on the scalp. The signal reflects both the intensity and pattern of neuronal activity, which depends on the brain's excitation

level and neuronal synchrony [62]. Different brain waves, characterized by specific frequency ranges, are associated with distinct physiological, cognitive, and emotional states. However, EEG is highly susceptible to artifacts from eye movements, muscle activity, and environmental noise, requiring proper preprocessing for reliable interpretation [190]. Clinically, EEG is widely used in neurology for diagnosing epilepsy and sleep disorders, while in research, it is essential for cognitive and affective neuroscience, mental health studies, and event-related potential analysis.

Respiratory Inductance Plethysmography (RIP) RIP tracks respiratory patterns, offering a non-invasive method for monitoring pulmonary function. The recording is performed using an elastic band with embedded coils, placed around the chest (rib cage) and/or abdominal walls. The output signal is proportional to the variation in the cross-sectional area of the surfaces enclosed by the bands [211]. Each band generates a respiration signal, from which metrics such as respiratory rate (the number of respiration cycles per minute) can be extracted. When two bands are used, the relationship between thoracic and abdominal respiration can be analyzed, including parameters like synchrony and correlation between the two signals. Changes in thoracoabdominal synchrony provide valuable information about physiological states, namely for local fatigue assessment [179].

Kinematic signals Kinematic signals encompass data related to movement, including acceleration, velocity, orientation, and position. They are commonly captured using Inertial Measurement Units (IMUs), which integrate accelerometers, gyroscopes, and magnetometers to track motion of body segments in three-dimensional space [60]. These signals are widely used in biomechanics, ergonomics, and human activity recognition, enabling applications such as posture analysis, gait assessment, and movement disorder monitoring.

2.1.2 Key Features

2.1.2.1 Physiological Meaning

Each type of biosignal tracks a specific physiological process of the body and offers information about the underlying activity. For this reason, these signals have physiological meaning and can play an important role in understanding complex systems and functions within the body.

2.1.2.2 Variability

Variability is an inherent and often desirable feature of biosignals, reflecting the dynamic nature of physiological systems. It is natural, predictable, and even positive for these signals to exhibit significant variability, both between different individuals (inter-subject variability) and within the same individual over time (intra-subject variability) [82]. This variability does not necessarily indicate pathological changes but rather the adaptability

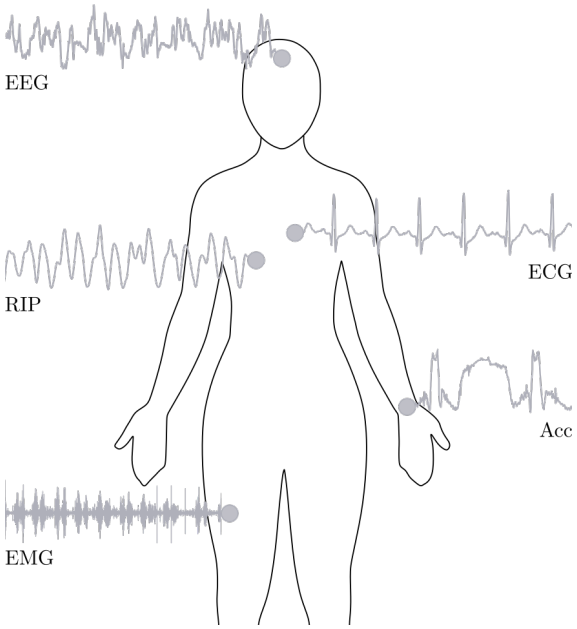


Figure 2.1: Illustrative examples of biosignals. From top to bottom: Electroencephalogram (EEG), Electrocardiogram (ECG), Respiratory Inductance Plethysmography (RIP), Accelerometry (Acc), and Electromyogram (EMG).

and responsiveness of the body to internal and external stimuli. For example, Heart Rate Variability (HRV) is a marker of the autonomic nervous system's adaptability [175]. However, this variability makes the interpretation of biosignals more complex, requiring robust analytical techniques to distinguish between normal physiological fluctuations and significant deviations.

2.1.2.3 Non-stationarity

Biosignals are inherently non-stationary, meaning their statistical properties, such as mean and variance, change over time [171]. This characteristic is due to the dynamic nature of physiological processes, which are influenced by factors like physical activity, emotional state, and circadian rhythms. For instance, the frequency and amplitude of EEG signals can vary significantly depending on whether a person is awake, drowsy, or asleep. Similarly, ECG waveforms can change with different levels of physical exertion. The non-stationarity of biosignals poses challenges for analysis, as many traditional signal processing techniques assume stationarity. Effective modeling and analysis of these signals often require methods that can adapt to changes in the signal characteristics over time [172].

2.1.2.4 Noise and Artifacts

Biosignals are frequently contaminated by noise and artifacts, which can obscure the underlying physiological information. Noise can arise from various sources, including environmental interference (e.g., baseline drift noise in ECG recordings), motion artifacts

(e.g., muscle contractions affecting EEG), and equipment imperfections. Artifacts are often non-physiological signals introduced by movement or other external factors, such as muscle contractions during EMG recordings or electrode displacements in EEG measurements. The presence of noise and artifacts complicates the analysis of biosignals, requiring preprocessing steps like filtering and signal enhancement to ensure that the relevant physiological information is retained. For example, in ECG signals, accurately detecting R-peaks is essential for computing key metrics such as HR and HRV. If noise interferes with peak detection, these metrics may become unreliable. Similarly, in anomaly detection or disease diagnosis, a corrupted signal makes it significantly harder to distinguish between true physiological disturbances and noise artifacts, potentially leading to misinterpretation.

2.1.2.5 Frequency Domain and Sampling Rates

The frequency content of biosignals varies widely, influencing the choice of appropriate sampling rates for accurate capture and analysis. For instance, ECG signals primarily contain frequencies between 0.5 and 40 Hz, while EEG signals span a broader range, often requiring sampling rates of 256 Hz or higher to capture high-frequency components. EMG signals can contain frequencies up to 500 Hz, requiring even higher sampling rates to avoid aliasing. Selecting a suitable sampling rate is crucial to preserving the integrity of the signal and ensuring that the relevant frequency components are accurately represented. The choice of sampling rate also impacts the storage and computational requirements of biosignal analysis, making it a critical consideration in both research and clinical applications.

2.1.3 Data Acquisition and Communication

The recording of biosignals relies on specialized sensors and devices designed to capture physiological processes with precision and reliability. These devices range from clinical-grade equipment used in controlled environments, such as hospitals and research laboratories, to portable and wearable sensors tailored for continuous monitoring in everyday settings. Depending on the type of biosignal being measured - electrical, mechanical, biochemical, or optical - different acquisition techniques and technologies are employed. For instance, ECG relies on electrodes placed on the skin to detect electrical activity of the heart, while EMG measures muscle activity through surface or intramuscular sensors. Similarly, respiratory signals can be captured using chest bands or airflow sensors, and brain activity is typically recorded using EEG electrodes. The choice of acquisition method significantly influences signal quality, resolution, and the type of information that can be extracted, ultimately shaping the subsequent processing and interpretation steps.

The way biosignals are acquired can introduce significant variability into the data, impacting their analysis and interpretation. Different acquisition methods, such as using portable devices versus clinical-grade equipment, can result in variations in signal quality

and the type of information captured. For example, an ECG measured with a one-lead wearable band might capture basic heart rhythms suitable for everyday monitoring, while a 12-lead ECG collected in a clinical setting provides a more comprehensive view of the heart's electrical activity.

Advancements in materials science, microelectronics, and nanotechnology have significantly driven the development of wearable biosensors, transforming them into key tools for continuous health monitoring and early anomaly detection. These devices leverage diverse sensing mechanisms - including mechanoelectric, optoelectronic, and ultrasonic technologies - to measure vital physiological parameters such as HR, blood pressure, and cardiac rhythm variations [131].

The integration of wireless communication technologies, such as Bluetooth, Near Field Communication (NFC), Wi-Fi, and Wireless Body Area Networks (WBAN), has further expanded the capabilities of these sensors, enabling real-time data transmission and remote health monitoring. One notable advancement in wearable biosensors is the development of textile-based wearables, where conductive materials, such as graphene, are embedded into fabrics. Graphene's exceptional electrical conductivity and flexibility offer high-fidelity signal acquisition, closely matching the performance of conventional gel-based ECG electrodes while improving user comfort and wearability [79].

Smartphones can also function as wearable devices, particularly for collecting kinematic signals. For example, they have been used to track posture in office workers throughout the day [141], offering the advantage of leveraging a device that most people already own.

Despite these innovations, several challenges remain. Ensuring stable and adhesive skin-electrode contact is crucial for minimizing signal artifacts and maintaining signal fidelity, particularly in motion-intensive applications. Additionally, as wearable devices become increasingly integrated into healthcare ecosystems, safeguarding patient data has become critical, necessitating robust encryption protocols to protect sensitive medical information. Moreover, the integration of Artificial Intelligence (AI) holds great potential for enhancing disease prediction and personalized health insights, yet optimizing AI models for real-time biosignal analysis remains an ongoing research challenge [122].

2.2 Applications

Biosignals provide access to detailed information about the body's functioning, most of the times without the need for any invasive procedures. Given their ability to reveal how the body's processes are evolving through time, they can consequently also reveal potential abnormalities. Hence, it is no surprise that they are widely used in clinical settings to assess health conditions and detect signs of diseases. However, as previously discussed, their usefulness is not limited to identifying comorbidities, and we can extract important observations from them in other contexts.

For instance, biosignals can be used to identify specific physiological states. One prominent example is stress detection, where wearable sensors track ECG, EEG, skin conductance, and other physiological markers to identify stress peaks and provide real-time feedback [21, 132]. This information can help individuals engage in stress-reducing strategies, such as breathing exercises or relaxation techniques. Another application is fall detection, particularly for elderly individuals or those with mobility impairments. Wearable sensors and accelerometers can detect sudden changes in posture or impact forces, triggering alerts for caregivers or emergency services [138]. Additionally, gait analysis using biosignals can help predict and prevent falls before they occur.

Beyond physiological state assessment, biosignals are also employed in biometrics and cybersecurity [178]. Unique physiological characteristics, such as an individual's ECG or EEG patterns, can be used as biometric authentication methods to enhance security systems. These methods offer a more secure and continuous authentication process compared to traditional passwords or fingerprint scans.

In human-computer interaction, particularly in brain-computer interfaces (BCIs), EEG-based systems enable users to control external devices, such as computers, wheelchairs, or robotic arms, using only neural activity, facilitating accessibility for individuals with motor impairments [177]. EMG and EEG signals are widely used in robotic prosthetics, allowing individuals with limb loss to intuitively control prosthetic limbs by translating muscle or brain activity into movement, significantly improving mobility and quality of life [33]. In sports, biosignals such as ECG and EMG provide insights into athlete fatigue and reaction times, allowing for personalized training regimens that enhance performance while minimizing the risk of injury[173].

2.2.1 Relevance in the field of Occupational Health

Data from the European Union shows that among individuals aged 20 to 64, the average working week in their main occupation is 36.4 hours, though working hours can be considerably longer [44]. Prolonged working hours and and extended exposure to occupational risk factors - such as awkward postures, repetitive movements, work pace, and noise - pose significant health and safety hazards, increasing the risk of occupational diseases [38]. According to the Protocol to the Occupational Safety and Health Convention, an occupational disease is any disease resulting from exposure to risk factors arising from work activity [77].

Biosignals monitoring through the use of werable devices can play an important role in preventing occupational diseases and mitigating work-related hazards. Physiological monitoring has emerged as a valuable tool in occupational settings, offering real-time insights into workers' physiological responses to various job-related stressors. Wearable sensors enable continuous tracking of key physiological parameters such as HR, respiratory rate, skin temperature, and physical activity levels, providing objective data on workers' physical and mental strain. This data is particularly relevant for occupations exposed

to extreme conditions, such as firefighters, miners, and construction workers, where monitoring heat stress, fatigue, and cardiovascular load (CVL) can help prevent work-related diseases and accidents. Additionally, physiological monitoring supports early detection of potential health risks, allowing for timely interventions, improved workload management, and personalized health recommendations [25].

In manufacturing, physical fatigue is a challenging safety problem since it increases the incidence of accidents and contributes to the development of musculoskeletal disorders [137]. In [119], the authors employed wearable sensors to detect whole-body fatigue in simulated manufacturing tasks, using IMUs and a HR tracker to estimate fatigue levels over time. To address the high prevalence of back injuries - one of the most common comorbidities in manufacturing - [54] proposed a real-time muscle fatigue measurement framework. This system utilizes EMG wireless sensors and a feature-based processing pipeline to monitor local muscle fatigue through signal characteristics such as Power Spectral Density (PSD) and mean frequency. While the results showed an evolution in fatigue levels over time, the study was conducted on only three subjects, highlighting the need for further validation.

Fatigue monitoring in occupational settings should be seen as an important aspect of workplace safety and productivity, as fatigue impairs cognitive and motor performance, increasing the risk of accidents and injuries. Wearable sensors offer a promising approach for real-time fatigue detection, enabling continuous monitoring in dynamic work environments. Various physiological signals, such as EMG, ECG, and motion-based metrics, have been used to assess fatigue levels, particularly in physically demanding occupations. However, while wearable-based monitoring has demonstrated significant potential, current models still face challenges related to data quality, generalizability across different workers, and real-world applicability. Addressing these limitations could lead to more reliable fatigue monitoring systems, ultimately enhancing worker well-being and informing occupational health policies [120].

Cardiovascular disease (CVD) remains the leading cause of all deaths worldwide, being responsible for an estimated 17.9 million mortalities per year [27]. Heavy physical work, awkward postures, repetitive movements, long working hours and overtime are just some of the many occupational factors associated with an increased risk of CVD [76, 92, 95]. In the following section, we will delve into the assessment of cardiovascular load in the workplace, with the aim of mitigating occupational risk for the worker, through the collection of biosignals and biosignal-derived metrics.

2.3 Cardiovascular Load Assessment in the workplace[1]

The field of ergonomics and occupational health and prevention of work-related disease considers a broad range of factors that impact physical strain at work [140]. These include environmental factors, job-related factors, task-related factors, and individual worker-related factors.

A range of objective and subjective metrics is available to quantify cardiovascular load (CVL). Objective metrics are based on the measurement of physiological variables (e.g., heart rate, oxygen consumption, blood pressure) and subjective metrics rely on the use of self-report instruments (or scales) to assess workers' perceived and experienced occupational exposure. However, there is inconsistent evidence linking occupational risk factors to CVD, highlighting a gap in knowledge. For that reason, the aim of the developed study was to identify how CVL is assessed in the workplace and to bring together related evidence-based recommendations for preventative measures. Specifically, we aimed to systematically search the Google Scholar database for observational studies in order to a) gather metrics used in these studies to assess CVL, b) summarize the related risk factors that were investigated, c) report the occupational groups and activities that these studies targeted, and d) summarize general recommendations for preventative measures for occupational health that resulted from these studies. A systematic search was conducted using Google Scholar for relevant literature up to June 22, 2022, following PRISMA guidelines [133]. The search focused on English language publications from peer reviewed scientific journals, irrespective of country of origin. The search terms for the database query were: "cardiovascular load" AND ("occupational" OR "workers") and the study selection process resulted in 76 articles.

In this section, we present an overview of the main results and findings.

2.3.1 Cardiovascular Load Assessment Metrics

The first aim of this study was to identify and gather metrics for the assessment of CVL that have been used in observational studies of occupational work. In Table 2.1, a brief description and the main limitations of each metric is presented.

The majority of the reviewed studies reported the use of objective metrics, in particular, HR-based metrics. HR-based metrics include the HR itself, the percentage in relation to the maximum HR ($\%HR_{max}$), Relative Heart Rate (RHR), Cardiovascular Load (CVL), Cardiovascular Strain (CVS), and HRV variables. One important characteristic about the measurement of HR that makes it the preferred metric for many researchers in the occupational context is that it is easy to monitor without disturbing the normal performance

[1]This section is based on the publication "Cardiovascular load assessment in the workplace: A systematic review" by **Mariana Dias**, Luís Silva, Duarte Folgado, Maria Lua Nunes, Cátia Cepeda, Marcus Cheetham, and Hugo Gamboa, published in the journal International Journal of Industrial Ergonomics (2023). In the present document, only a summary of the most relevant aspects of the study is provided; for a comprehensive overview, please refer to the open-access publication.

of the work. However, HR does not depend exclusively on the physical demand of the tasks that are being performed - it depends on various factors, such as individual traits, namely the stroke volume. For instance, athletes or trained subjects have lower HR as a consequence of a higher stroke volume [85]. As such, looking directly at the values of HR can be misleading an that is why HR-based metrics are used.

Table 2.1: Description and main limitations of the objective metrics for CVL assessment.

Metric	**Description**	**Limitations**
HR	Number of heart beats per minute. Feasible continuous and direct monitoring during work.	Does not consider inter-individual variability (e.g. stroke volume).
$\%HR_{max}$	Intensity of the worker's HR in relation to the HR_{max}.	Does not consider HR_{rest}, not representing load above the resting state.
RHR	Intensity of effort in relation to HRR. Computed from the direct measure of HR and HR_{rest} and the estimation of HR_{max}.	-
%CVL	Relation between HR_{work} (subtracting HR_{rest}) and $HR_{(8h)}$ (maximum acceptable HR for an 8-hour shift). Easy to monitor.	Interpretation is not intuitive. There is lack of validated scales.
%CVS	Relation between HR_{work} (subtracting HR_{rest}) and HR_{rest} itself.	Few studies have used it. Lack of validation. Interpretation is not intuitive.
VO_2	Measurement of oxygen consumption during work. Good measure to objectively describe stress.	Expensive equipment required. Continuous monitoring is often not feasible during work.
RVO_2	Measurement of oxygen consumption during work, considering individual VO_{2min} and VO_{2max}.	Expensive equipment required. Continuous monitoring is often not feasible during work. Requires an estimation of VO_{2max}.
BP	Pressure of blood in the arteries. High blood pressure is a powerful predisposing risk factor for CVD. Highly studied parameter.	Difficult to monitor in occupational context. Requires the worker to stop momentarily.
HRV	Variation of the period between consecutive heartbeats. May contain relevant indicators of cardiac impairments.	Requires the use of devices that detect the time fluctuations between consecutive heart beats (e.g. ECG) and computer-based analytical tools.
ECG	Record of the electrical activity of the heart. HRV and relevant information can be extracted from the ECG.	Requires analytical tools or a clinician to interpret. The ECG signal obtained during work is easily corruptible with noise.
BLA	Blood Lactate concentration, a parameter commonly used in clinical context, which increases exponentially in intense exercise.	Few studies have used it in occupational context. Continuous monitoring not feasible. Highly intrusive.
PSI	Measure of heat stress, based on rectal temperature and HR.	Does not measure cardiovascular constraints directly. Lack of validation studies on applicability in occupational context.

The most frequently used metric was the RHR. It uses the HR at work while accounting for resting HR and compares it to the Heart Rate Reserve (HRR) - the difference

between maximum HR and resting HR (which depend on factors such as age and stroke volume) [116]. It is an intuitive metric because it expresses intensity as a proportion of an individual's available heart rate range, making it more personalized than absolute heart rate measurements and better suited for comparing effort across individuals. Oxygen consumption (VO_2) is also used as a cardiovascular metric [143] but it is much more difficult to measure, particularly in the workplace, once it requires measuring the air flow rates during breathing and the worker must be wearing a mask and it is very likely to interfere with the normal performance of the job.

As previously mentioned, from the ECG signal it is possible to extract meaningful information, such as the HRV (from the distance between consecutive R peaks) and cardiac anomalies (from changes in the different segments of the signal). Although only one study used this approach (to count the occurrences of extrasystoles during work [187]), we believe it is a potential focus of future research, since it is widely used in the clinical context and has proven to be very informative. Furthermore, the number of automatic algorithms for ECG processing is increasing, thus sparing laborious manual data analysis [104]. Other metrics include the Blood Pressure (BP), Blood Lactate Accumulation (BLA) [143] and the Physiological Strain Index (PSI) [40].

2.3.2 Risk Factors and Occupational groups

The second aim of this study was to summarize the risk factors investigated in the observational studies. A wide range of different aspects were reported to have an influence on cardiovascular stress:

- **Environment**: Exposure to very high or low temperatures, humidity, noise, and uneven terrain.
- **Job Characteristics**: Working for a prolonged number of hours, high physical demands.
- **Task-Specific Factors**: Handling heavy loads, performing repetitive tasks, and postural strain.
- **Individual Factors**: In cases of mismatch between the workers physical characteristics (age, body mass index, fitness level, and health conditions) and job requirements.

For this multi-factorial phenomenon, the relevance of each factor depends on the occupational context. Regarding the occupational groups and activities that the observational studies targeted, most studies focused on occupational health in the manufacturing industry [145, 189], construction [117, 8], agriculture and forestry [165, 11, 16] sectors. These occupations employ large parts of the work force and are known to be associated with the emergence of occupational diseases. Nevertheless, less physically demanding and/or sedentary jobs are also known to have negative repercussions for the health of the worker. We found little research has been conducted in this sector of activity.

2.3.3 Main Findings and Recommendations

Physical activity can both benefit and stress the cardiovascular system. While the literature suggests that moderate and moderate-to-high physical activity is beneficial for promoting cardiovascular health, excessive occupational physical activity can lead to chronic stress, increasing CVD risk [69, 106]. One thing to consider, is that heat stress exacerbates cardiovascular strain, particularly in outdoor jobs - hence, in these situations, particular attention must be payed to high levels of physical activity. Moreover, findings suggest that balancing workload intensity and recovery time is critical for worker health.

Despite the extensive body of research on occupational health, evidence-based recommendations remain limited. However, several general guidelines have emerged to help mitigate cardiovascular strain and enhance worker well-being. Workload management strategies should include implementing workload thresholds, such as maintaining a %CVL below 40%, and ensuring adequate rest periods to prevent excessive physiological stress [165]. Workplace design improvements can further reduce strain through the introduction of ergonomic tools, automation, and strategic task rotation to minimize repetitive movements. The use of wearable technology offers a promising approach to real-time health monitoring, allowing for continuous tracking of HR-derived metrics to identify high-risk workers and integrating physiological sensors into workplace health programs [25]. Finally, occupational health policies should establish clear guidelines for acceptable cardiovascular load based on job demands and promote regular health screenings for employees in high-risk roles, fostering a proactive approach to workplace well-being.

2.3.4 Ethical considerations

In the workplace, ensuring the protection of workers is essential. This includes guarantees for employment security, stringent data protection policies to safeguard personal health information, and clear, ethical guidelines to prevent discrimination based on biosignal data. Additionally, measures must be in place to maintain confidentiality, define clear consent protocols, and establish standards to ensure that data is used solely for health improvement purposes. Lastly, in public health contexts, oversight is required to prevent misuse of biosignal data by private entities and to ensure that individuals retain control over their health information.

2.4 Final Remarks

Biosignals provide a powerful means of accessing and analyzing physiological processes, enabling valuable insights into human health and performance. Their applications span across diverse domains, including clinical diagnostics, occupational health, sports science, and human-computer interaction, where they serve as critical tools. The increasing accessibility of biosignal acquisition through wearable devices has further expanded their

potential, allowing real-time tracking of physiological states outside controlled laboratory or clinical environments.

Despite these advantages, biosignal processing presents numerous challenges. For instance, signal contamination due to motion artifacts or other noise sources can significantly impact the reliability of analyses. Moreover, biosignals are inherently non-stationary and subject to physiological variations both within and across individuals, making robust interpretation difficult. These challenges necessitate advanced processing techniques capable of effectively filtering noise, extracting relevant features, and adapting to different acquisition conditions.

The development of more sophisticated, scalable, and generalizable processing methods is crucial for maximizing the utility of biosignals. The next chapter explores the role of Deep Learning in addressing these challenges, leveraging data-driven approaches to improve biosignal processing, enhance generalization, and optimize computational efficiency.

3

Deep Learning: Foundations, Challenges, and Application in Biosignals

Over the past decade, Deep Learning (DL) has emerged as the dominant and most effective Artificial Intelligence (AI) paradigm across various application fields. In biosignal processing, it has demonstrated significant potential, and this work aims to build upon that progress.

This chapter begins by introducing the core theoretical concepts of DL that form the foundation of this thesis. Next, it addresses one of the main challenges of DL - its high resource demands - and explores emerging strategies to mitigate these limitations. Finally, it presents a narrative review of the literature on DL applications in biosignal processing, which serves as the basis for this research.

3.1 Fundamentals

3.1.1 Artificial Intelligence, Machine Learning, Deep Learning

The concept of intelligence has been approached from multiple perspectives - philosophical, psychological, biological - each emphasizing different aspects. While not universal, a simple definition states that intelligence is "the ability to acquire and apply knowledge and skills" [102]. However, some authors argue that it is not possible to define intelligence without including broader aspects, such as "sensation, perception, association, memory, imagination, discrimination, judgement and reasoning" [186, 102]. Inevitably, the concept of Artificial Intelligence inherited some of this controversy, and no universally accepted definition exists [2]. François Chollet provides a straightforward definition, describing AI as "the effort to automate intellectual tasks normally performed by humans" [34].

Machine Learning (ML) is a subfield of AI that focuses on algorithms that do not rely on explicitly programmed rules but instead use statistical methods to estimate complex functions [55]. DL, in turn, is a subfield of ML that leverages deep neural networks

to learn hierarchical representations of data. Unlike traditional ML algorithms, which often require manually engineered features, DL models can automatically extract relevant features from raw data such as images, time series, and videos, making them particularly powerful for learning complex patterns [81].

3.1.2 From the Artificial Neuron to the Artificial Neural Network

The term "neural networks" originates from the inspiration behind these computational models: the biological neurons. In a highly simplified view, biological neurons receive electrical stimuli through synapses and, when a neuron receives a sufficiently strong signal, its membrane depolarizes, generating an action potential, which is then propagated to subsequent neurons. This mechanism is behind the transmission of information across the nervous system and Artificial Neural Networks (ANNs) drew inspiration from it [81]. Early artificial neurons were designed to perform basic logical operations, using binary inputs and outputs (0 or 1). Over time, the concept evolved into more sophisticated models where a neuron receives multiple inputs, each associated with a weight. These inputs are combined through a weighted sum, and the result is processed through a function known as the activation function, f, which determines the neuron's output, as represented in Figure 3.1 [55]. The activation function is a mathematical function applied to the output of a neuron that determines whether and to what extent the neuron should be activated and the most common types include the sigmoid function, the hyperbolic tangent (tanh), the rectified linear unit (ReLU), and the softmax [129].

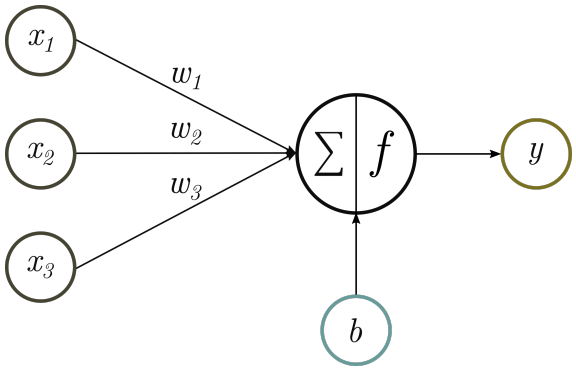


Figure 3.1: Schematic representation of an artificial neuron. x_i and w_i are the inputs and respective weights, b stands for the bias term (activation threshold), and y is the ouput.

ANNs are structured collections of artificial neurons (also referred to as nodes) arranged in layers. The first one to be developed was the Feed Forward Neural Network (FFNN), represented in Figure 3.2. The connections between neurons are associated with adjustable weights, which are optimized through learning algorithms [176]. The introduction of non-linearity - by using activation functions and stacking multiple layers - marked a significant shift in neural networks, enabling them to approximate complex functions beyond simple linear mappings [70]. This ability to learn hierarchical representations is what makes deep neural networks particularly powerful for modern AI applications.

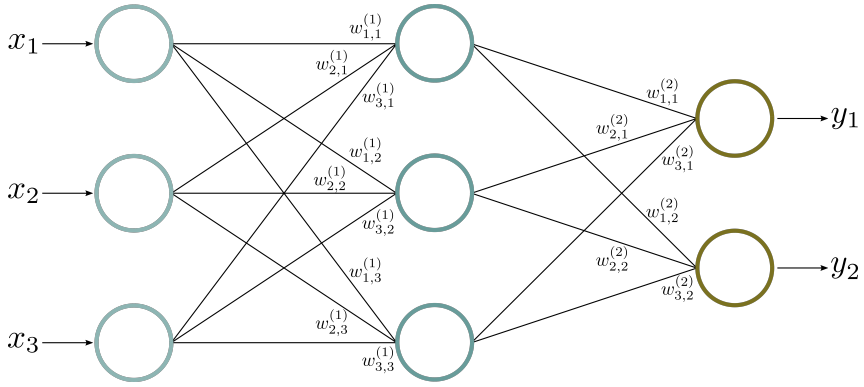


Figure 3.2: Schematic representation of a Feed Forward Neural Network (FFNN). x_i and y_j represent the inputs and outputs of the network. $w^{(l)}_{i,j}$ denotes the connection weight between neuron i in layer l and neuron j in layer $l+1$.

3.1.3 How do Artificial Neural Networks learn?

The training phase is when the learning takes place. In order to train an ANN, just like any other ML model, it is necessary to have access to a training set. In simple terms, the ML system is presented with many examples relevant to a task and finds the statistical structure that determines the rules behind this task [88]. Machine Learning paradigms can be generally divided in 3 main categories: (i) supervised learning, where the training examples provide both input data and the corresponding correct outputs (labels); (ii) unsupervised data, where we only have access to input data without explicit labels and the model is meant to identify patterns within the data; (iii) reinforcement learning, where an agent interacts with an environment and learns through a system of rewards and penalties [51]. Additionally, falling out of these categories, there are two others: semi-supervised learning, which combines a small amount of labeled data with a larger pool of unlabeled data to improve learning efficiency, and Self-Supervised Learning (SSL), a variant of supervised learning in which the model generates its own supervisory signals from the data, often by solving pretext tasks before fine-tuning on a specific objective.

For explaining the learning process, we will use the example of an FFNN in a supervised learning scenario. For each training sample, an associated expected output is provided and a loss function is used to compute the training error, which compares the algorithm's output to the expected one. The main idea of the training process is to reduce this error, making the network's output as close to the correct labels as possible [55].

Gradient descent is an optimization technique used to minimize the training error, by iteratively adjusting the parameters of the model (weights and biases) in order to find those that lead to the lowest error. For that, the parameters are adjusted in the opposite direction of the partial derivatives of the loss function towards its minimum. It is often necessary to optimize complex functions that may have many local minima or that have multidimensional inputs - this makes optimization difficult and, therefore, it is usual to define a stopping criteria different from finding the global minimum. Furthermore, in

DL, the objective is not necessarily to reach the global minimum of the loss function, as doing so could lead to a model overfitted to the training data. This phenomena will be discussed further ahead in section 3.1.5.

The gradient of the loss function can be computed in different ways during training: it can be calculated using the entire training set at each iteration (batch gradient descent), using small random subsets of the training set (mini-batch gradient descent), or using a single training example at a time (stochastic gradient descent, SGD) [158]. An epoch refers to one complete pass through the entire training dataset. Given a dataset with N samples: in batch gradient descent, the model parameters are updated once per epoch; in SCG, the parameters are updated N times per epoch and in mini-batch gradient descent, with a batch size of n, the parameters are updated N/n times per epoch.

The gradient descent method uses the gradient of the loss function (with respect to the weights) to make a step change in w to lead it towards the minimum of the error curve. This is an iterative method, where the weights and biases are updated according to the expressions 3.1 and 3.2 [158].

$$w_{ij}^{(l)} = w_{ij}^{(l)} - \alpha \frac{\partial L(w,b)}{\partial w_{ij}^{(l)}} \tag{3.1}$$

$$b_{ij}^{(l)} = b_{ij}^{(l)} - \alpha \frac{\partial L(w,b)}{\partial b_{ij}^{(l)}} \tag{3.2}$$

Where α is the learning rate and L is the loss function. The learning rate is a hyper-parameter which controls how much to change the value of the weight in the opposite direction from the gradient. Choosing an appropriate value for α can be challenging: if it is too small, the optimization process will take many iterations to reach the minimum loss and may get stuck in a local minimum; if it is too large, the optimization may diverge [88].

Several algorithms have been developed to address this issue. One example is the addition of a momentum term [151], which accelerates gradient descent in the relevant direction by incorporating a fraction of the update vector from the previous time step into the current update. This approach helps achieve faster convergence while reducing oscillations. The Adaptive Moment Estimation (Adam) optimizer [91] builds on this idea by combining the benefits of momentum-based optimization with adaptive learning rates. It maintains an exponentially decaying average of past gradients m_t (similar to momentum) and an exponentially decaying average of past squared gradients v_t (used for adaptive step sizes) [129]. The parameter θ update rule is given by:

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{3.3}$$

where $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected moment estimates:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \qquad m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_\theta L(\theta) \tag{3.4}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}, \qquad v_t = \beta_2 v_{t-1} + (1-\beta_2)(\nabla_\theta L(\theta))^2 \tag{3.5}$$

When working with neural networks, the most commonly used method to compute these gradients is the backpropagation method [159], an optimizer which uses the chain rule of differentiation for applying the gradient descent to the equations that define the network.

3.1.3.1 Data as the key for a good model

The quality and organization of the dataset play a crucial role in the success of any DL model. To ensure a fair and reliable evaluation, the available data should be systematically divided into three distinct subsets: training, validation, and testing. The training set is used to update the model's parameters, allowing it to learn patterns and relationships within the data. The validation set consists of examples not seen during training and is used to monitor the model's performance through epochs. By assessing the loss on validation data, we ensure that the model generalizes well to previously unseen examples rather than memorizing the training data: when the validation loss starts increasing, it means that the model is overfitting to the training data. Finally, the test set provides an unbiased evaluation of the model's final performance on completely new data, simulating real-world deployment scenarios.

A critical concern in this process is data leakage, which occurs when information from the test or validation set inadvertently influences the training process. Data leakage can lead to overly optimistic performance estimates that do not reflect the model's true ability to generalize. To avoid this, it is essential to carefully separate the datasets and ensure that no pre-processing steps introduce unintended correlations between them. Proper dataset partitioning and rigorous evaluation strategies are fundamental to building robust models.

Particularly in biosignal datasets, the representativity of data often suffers from imbalances, particularly in terms of the distribution of healthy versus unhealthy samples. In many cases, datasets contain a higher proportion of data from healthy individuals compared to those with specific conditions, such as rare diseases. This imbalance can lead to biased models that perform well on majority classes but poorly on underrepresented conditions. Addressing these biases during both data collection and model training is crucial for ensuring accurate and generalizable outcomes in biosignal analysis.

3.1.4 Architecture Paradigms

FFNNs were the first type of neural networks but since its development, a diverse set of architectures have been implemented, having reached much more complex and detailed structures. These architectures vary in their design, strenghts, types of data that they are suited for, and applications. In this section, the three DL architectural paradigms relevant

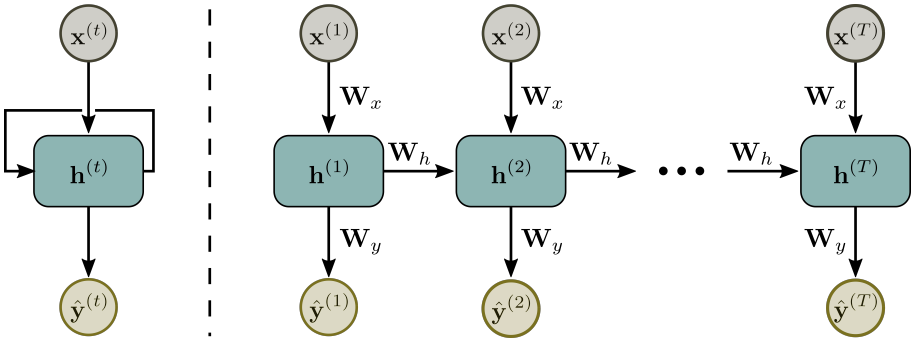


Figure 3.3: Schematic representation of a Recurrent Neural Network (RNN). $x^{(t)}$ and $\hat{y}^{(t)}$ represent the input and corresponding output at time step t, respectively. W_x, W_h, and W_y denote the weight matrices for the input, recurrent (hidden state), and output connections.

to understand the developed work are explored: Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Transformers.

3.1.4.1 Recurrent Neural Networks (RNNs)

RNNs are neural networks that were designed for sequence analysis. Sequences are made of data points that are sequentially dependent, meaning that a data point at a specific time step is affected by previous points and might influence subsequent points [166]. To model the sequential dependencies between time steps, RNNs include feedback connections among hidden units, which allow a "memory" of previous inputs to persist in the network's internal state [159]. RNNs are also trained with backpropagation and the forward pass of an RNN is identical to that of a feedforward network, except that the hidden layers receive as inputs both the current external input and the output from the previous timestep [57], as represented in figure 3.3.

Considering an RNN with I input nodes, a single hidden layer (with H hidden nodes) and an output layer (with K output nodes), which receives as input a sequence x, the forward pass is as represented in expressions 3.6 and 3.7 [57]. In the equations, a_h^t represents the activation of hidden unit h at time step t, computed as the weighted sum of the current input x_i^t and the activations from the previous time step $o_{h'}^{t-1}$. The weight w_{ih} denotes the connection weight between input node i and hidden node h, while $w_{h'h}$ represents the recurrent weight connecting hidden node h' from the previous time step to hidden node h at the current step. The hidden activation a_h^t is then passed through a non-linear activation function f, yielding the output o_h^t, which serves as the hidden state for the next time step or as input to the output layer.

$$a_h^t = \sum_{i=1}^{I} w_{ih} x_i^t + \sum_{h'=1}^{H} w_{h'h} o_{h'}^{t-1} \tag{3.6}$$

$$o_h^t = f(a_h^t) \tag{3.7}$$

Similarly to what is done in the learning phase when working with FFNN, it is necessary to compute the partial derivatives of the loss function with respect to the weights. The most used algorithm for efficiently calculate these derivatives is backpropagation through time (BPTT) [210], whose equations are represented in expressions 3.8 and 3.9, where δ denotes the backpropagated error term [57]. It is important to notice that the same weights are used every timestep.

$$\frac{\partial L}{\partial w_{ij}} = \sum_{t=1}^{T} \delta_j^t \times \frac{\partial a_j^t}{\partial w_{ij}} = \sum_{t=1}^{T} \delta_j^t o_i^t \tag{3.8}$$

$$\text{with } \delta_h^t = f'(a_h^t)(\sum_{k=1}^{K} \delta_k^t w_{hk} + \sum_{h'=1}^{H} \delta_{h'}^{t+1} w_{hh'}) \tag{3.9}$$

However, when trying to model long-term dependencies with basic RNN models, the gradients calculated for updating the network's weights during the back-propagation step, tend to vanish [17, 66]. Recurrent networks involve the chain multiplication of the derivative of the activation function, once per time step, and gradients propagated over many timesteps tend to either vanish or explode (being the first phenomenon much more probable than the second) [55].

Long Short Term Memory (LSTM) Long Short Term Memory (LSTM) is a type of RNN with a gated structure, which enables it to handle long input sequences [67]. The central idea behind the LSTM architecture is a memory cell which can maintain its state over time, containing non-linear gating units to regulate the information flow into and out of the cell: the input, output and forget gates [58].

Figure 3.6 illustrates an LSTM cell and the expressions from 3.10 to 3.15 are the equations for the forward pass. Having as input a sequence x, x_t represents the input at the current timestep t and h_{t-1} and C_{t-1} represent, respectively, the output and the cell state from the previous timestep. W, b are the weights and biases and the indexes f, i and o correspond to the forget, input and output gates, respectively. These multiplicative gates are sigmoid layers and each has a different task. The forget gate (3.10) takes as input x_t and h_{t-1} and it is what enables the cell state to be reset, as its output will multiply the previous cell state C_{t-1} (see 3.13). The input gate (3.11) "decides" which values will be updated and from 3.12 the new values to add to the cell state are computed. Finally, in the output gate (3.14) it is decided what part of the current cell state is going to be output (3.15).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (forget\ gate) \tag{3.10}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (input\ gate) \tag{3.11}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (candidate\ values) \tag{3.12}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (cell\ state) \tag{3.13}$$

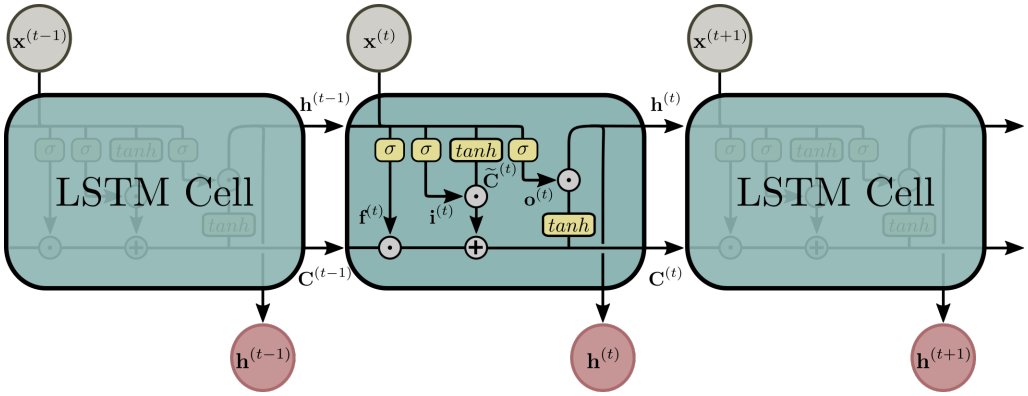


Figure 3.4: Schematic representation of a Long Short Term Memory (LSTM) cell. $x^{(t)}$, $h^{(t)}$, and $C^{(t)}$ represent the input, output, and cell state at timestep t, respectively. f, i, and o correspond to the forget, input, and output gates. σ denotes sigmoid activation layers, and $tanh$ denotes hyperbolic tangent activations. The dot symbol inside a circle ($\odot$) represents the element-wise (Hadamard) product, which corresponds to the * symbol used in the equations throughout this chapter.

$$o_t = \sigma(\, W_o \cdot [\, h_{t-1},\, x_t] + \, b_o) \quad (output\ gate) \tag{3.14}$$

$$h_t = \, o_t * tanh(\, C_t) \quad (output) \tag{3.15}$$

The gates allow LSTM cells to store and access information over long periods of time, mitigating the vanishing gradient problem. For example, as long as the input gate remains closed (i.e. has an activation near 0), the activation of the cell will not be overwritten by the new inputs arriving in the network, and can therefore be made available to the net much later in the sequence, by opening the output gate [57].

Gated Recurrent Unit (GRU) A Gated Recurrent Unit (GRU) is a more recently developed type of RNN with a gated architecture. The main differences from the LSTM are that it uses two gates (the update and reset gates) instead of three and that it doesn't use the cell state to transfer information, but rather the hidden state [36]. It can be thought of as a modification of the LSTM with a less complex architecture and, consequently, more computationally efficient [88].

Figure 3.5 illustrates the structure of the GRU and its components. The update gate, represented by z_t, "couples" the forget and input gates from the LSTM architecture into one, which simultaneously controls how much of the previous memory content (h_{t-1}) to forget and how much of the new content (x_t) is to be added, through the computation of expression 3.16. The reset gate, r_t, allows the unit to forget the previous hidden states and it is computed as in expression 3.17. The candidate hidden state ($\tilde{h}_t$) is done similarly to that of the the LSTM (expression 3.18) [37]. Finally, at timestep t the state of the GRU is the linear interpolation between the previous activation (h_{t-1}) and the candidate hidden state ($\tilde{h}_t$) [36].

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \quad (update\ gate) \tag{3.16}$$

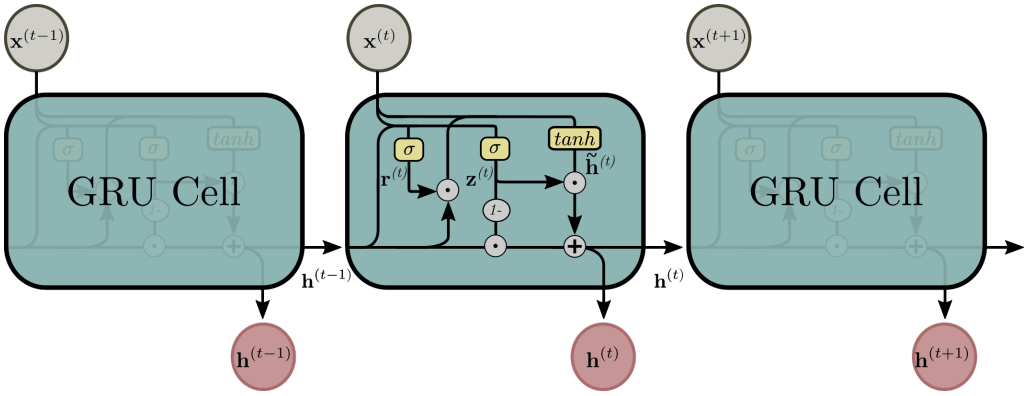


Figure 3.5: Schematic representation of a Gated Recurrent Unit (GRU) cell. $x^{(t)}$ and $h^{(t)}$ represent the input and output at timestep t, respectively. r and z correspond to the reset and update gates. σ denotes sigmoid activation layers, and $tanh$ denotes hyperbolic tangent activations. The dot symbol inside a circle ($\odot$) represents the element-wise (Hadamard) product, which corresponds to the * symbol used in the equations throughout this chapter.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \quad (reset\ gate) \tag{3.17}$$

$$\tilde{h}_t = tanh(W \cdot [h_{t-1} * r_t, x_t] + b_h) \quad (candidate\ activation) \tag{3.18}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (output) \tag{3.19}$$

Despite its simpler architecture and the fact that it may not have as much representational power as LSTM, it has been evidenced that it can outperform LSTM networks [36], particularly on smaller datasets.

GRU and any other type of RNN can be implemented in a bidirectional fashion, where two GRU layers process the sequence in both forward and backward directions before merging their outputs, as represented in figure 3.6. This bidirectionality allows the network to incorporate both past and future context when making predictions, potentially enhancing their ability to capture long-range dependencies in sequential data.

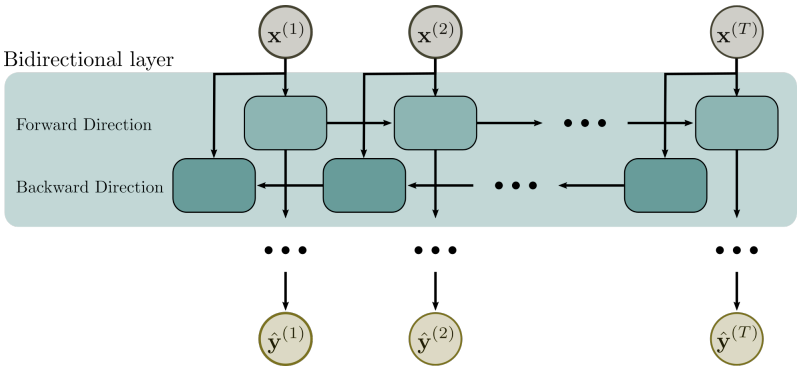


Figure 3.6: Schematic representation of bidirectionality in an RNN.

3.1.4.2 Convolutional Neural Networks (CNNs)

CNNs were originally designed for image processing, for tasks such as image classification, object detection, and image segmentation. They utilize convolutional layers to automatically learn spatial features from input data, including edges, textures, and shapes. Each convolutional layer applies a set of learnable filters, called kernels, to the input, producing a feature map that highlights relevant patterns in the data [129]. The convolution operation is mathematically represented in Equation 3.20, where $f_k(x, y)$ denotes the feature map, x the input image, $w_{i,j}$ the filter weights, and b the bias term.

$$f_k(x, y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} w_{i,j} x_{(i+x),(j+y)} + b \tag{3.20}$$

To improve computational efficiency and prevent overfitting, CNN architectures typically include pooling layers between convolutional layers. Pooling layers reduce the dimensionality of feature maps while preserving the most significant information. The pooling operation involves selecting the maximum (or alternatively, the mean or minimum) value from each patch of the feature map, as represented in Equation 3.21.

$$p(x, y) = \max_{i=0}^{m-1} \max_{j=0}^{n-1} f(x + i, y + j) \tag{3.21}$$

Although CNNs were initially designed for computer vision, they have demonstrated effectiveness in time-series analysis due to their ability to extract local patterns through convolutional filters. In biosignal processing, 1D CNNs have been used to extract spatial-temporal features in signals such as ECG, EEG, and EMG [101]. Unlike traditional 2D CNNs, which process spatial features in images by applying 2D convolutional filters, 1D CNNs operate along a single temporal dimension, making them particularly suited for sequential data such as biosignals. In 1D CNNs, the convolutional filters slide across the time axis of the input signal, capturing local dependencies within the sequence rather than spatial relationships. This enables the model to extract temporal patterns and short-term dependencies effectively. Their hierarchical feature-learning ability makes them particularly useful for tasks where local dependencies play a crucial role in understanding the underlying patterns in the data.

3.1.4.3 Transformers

Transformers, introduced by Vaswani et al. in 2017 [202], are a class of DL architectures that relies entirely on self-attention mechanisms, eliminating the need for recurrence or convolutions. While RNNs process inputs sequentially, Transformers employ self-attention, allowing for efficient parallelization and long-range dependency modeling. At the core of the Transformer is an encoder-decoder architecture, where both components are composed of multiple layers of self-attention and position-wise feed-forward networks. Each token in a sequence attends to all other tokens, enabling the model to capture

relationships across the entire input. Initially introduced for Natural Language Processing (NLP), transformers have since been adapted to various domains, including time-series analysis, biosignal processing, and computer vision [10, 129]. In biosignal processing, a token can represent a single timestep, a segment of a signal, or a feature extracted from the raw data, depending on how the sequence is structured.

The core component of a transformer model is the self-attention mechanism, which allows each input element to attend to all other elements in the sequence, rather than being processed sequentially. Given an input sequence represented as a matrix $X \in \mathbb{R}^{n \times d}$, where n is the sequence length and d is the feature dimension, the self-attention mechanism computes three learnable matrices: the query Q, the key K, and the value V. These are obtained by multiplying the input X by corresponding weight matrices, such as in Equation 3.22, where $W^Q, W^K, W^V \in \mathbb{R}^{d \times d_k}$ are trainable matrices [202].

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V \tag{3.22}$$

The attention scores between each pair of tokens are computed using the scaled dot-product attention, which determines how much focus each token should place on others in the sequence. Its computation is described in Equation 3.23, where the denominator $\sqrt{d_k}$ is a scaling factor that prevents excessively large values in the softmax function, ensuring stable gradients during training [202].

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.23}$$

To improve expressiveness, transformers employ multi-head attention, where multiple attention layers operate in parallel, each learning different attention patterns. The outputs from these heads are concatenated and projected back into the model's feature space as in Equation 3.24, where each head follows the same self-attention operation with independent learned parameters.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \tag{3.24}$$

Transformers also introduce positional encodings to compensate for the lack of sequential processing inherent in their architecture. Since self-attention treats inputs as an unordered set, positional encodings encode the token's position within the sequence, allowing the model to capture relative and absolute positional information. The transformer architecture is structured as a stack of identical layers, each containing multi-head self-attention followed by position-wise feedforward layers with residual connections and layer normalization. The feedforward layers apply transformations independently to each token using two linear layers with an activation function, as in Equation 3.25, where W_1, W_2, b_1, b_2 are learnable parameters.

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{3.25}$$

While the Transformer architecture is entirely built around self-attention, attention mechanisms can also be integrated into other DL models, such as RNNs and CNNs, to enhance their ability to capture long-range dependencies and focus on relevant information. In RNNs, attention layers allow the model to selectively attend to different parts of the input sequence at each decoding step, rather than relying solely on the final hidden state. This is achieved by computing attention scores between the current decoder state and each encoder hidden state, generating a context vector that dynamically influences the output. In CNNs, attention mechanisms can be incorporated to assign different importance levels to spatial regions in an image or time steps in a biosignal [129].

3.1.5 Complexity and Generalization

Generalization is a concept that preceded ML. Its original use was to state how well a statistical model performs on data that was not used for training the model [35]. As in statistics, the goal of an ML or DL model is not that it performs well on the training dataset but rather that it has good performance over new data, ie, that it generalizes well to previously unseen data [34].

Complexity, on the other hand, refers to a model's capacity, which is determined by the number of parameters it contains. A more complex model has more parameters that can be adjusted, allowing it to better fit highly variable data. However, higher complexity does not necessarily lead to higher generalization. While a more complex model will more easily adjust to the training data and achieve good training performance, it is also more prone to overfitting. Overfitting is the opposite of generalization and it occurs when a model achieves good performance in the training dataset but not on new data. The goal is to find the right balance where the model is complex enough to capture meaningful patterns but not so complex that it memorizes the training set instead of learning generalizable features [88].

In DL, when creating a new model, it is necessary to find an appropriate network size, i.e., number of layers and nodes. Usually, the correct approach is to start with small networks, with few units and parameters, and increase its complexity until the addition of parameters no longer adds representational power and the performance stops improving in the validation data. To prevent overfitting, several regularization techniques can be employed, *ie*, strategies to prevent the model from fitting the training data too closely. One method that is commonly used in DL is dropout, which consists of randomly and temporarily dropping out (by setting to zero) units along with their connections from the network during training [185]. This helps reducing the model's reliance on specific parameters, promoting robustness and improving generalization to unseen data.

A systematic literature review on computationally efficient DL models for detecting diabetic retinopathy (DR) investigated the trade-off between model accuracy and computational complexity. The authors show not only that higher complexity does not always

translate to better results (especially when the model is deployed in constrained environments) but they go even further, stating that models with fewer parameters tend to generalize better with less overfitting, making them more efficient in training and inference. Even in medical applications, while the search for high-accuracy models is desirable, the significant computational cost is overlooked, and this burden is particularly problematic in real-world clinical settings, where real-time performance, energy efficiency, and hardware limitations are crucial factors [65].

The traditional generalization concept assumes that the test data follows the same distribution as the training data, which is not always the case. Domain Generalization (DG) is a term used for defining the ability of ML models to perform well on unseen data whose distribution is different from the training set. While humans generalize naturally to novel scenarios, ML models often struggle due to their reliance on the assumption that training and testing data follow the same distribution (independent and identically distributed assumption) [207]. The domain shift problem occurs when the statistical properties of the data change between training (source domain) and testing (target domain), leading to significant performance drops in machine learning models. DG aims to improve model robustness against these shifts and there are several approaches for DG, such as Reinforcement Learning strategies, in settings where environment variations exist or Meta-Learning, simulating domain shifts during training [223]. The particularity of DG strategies is that they do not rely on target domain data for adaptation, distinguishing it from Domain Adaptation (DA), which assumes access to some target domain data. DA, as the name suggests, is a technique used to adapt a model trained on one domain (source) to perform well on a different domain (target). Unlike DG, the objective is not to create models that are generalizable to any domain *a priori* but rather to adapt them as the need arises [45].

In order for a model to generalize well in data that does not follow the same distribution as the training data, ie, to achieve DG, particularly in classification tasks, a much higher complexity is often necessary [103]. Highly complex models are disadvantageous in most cases for several reasons, such as the need for high-performance hardware, complex set ups, long training times, and the need for very big amounts of data. This topic leads us to one of the biggest challenges faced in DL: its resource demands and consequent environmental impact. The following section delves into this issue, discussing the growing concerns around DL's resource consumption and exploring emerging strategies for mitigating its impact. Notably, many of these approaches align more closely with DA rather than DG. While DA requires access to some target domain data, it offers a more computationally efficient alternative to DG, where the model must generalize without adaptation. Methods such as Transfer Learning (TL) capitalize on this trade-off, reducing the need for extensive retraining and improving efficiency while maintaining strong performance across domains.

3.2 Resource Demands: Challenges and Emerging Strategies

Artificial Intelligence is often viewed as a transformative solution, promising to solve some of the biggest challenges of modern society. But have we paused to consider whether, in solving these problems, AI might also be introducing new ones? The rapid rise of AI has undoubtedly brought remarkable advancements, yet it also raises ethical, environmental, and societal concerns that must not be overlooked. As we push forward with AI-driven approaches, it is essential to reflect on the potential trade-offs. Particularly when applying them to a sensitive field like healthcare, there are several aspects that need to be addressed, such as potential bias in training data that can lead to disparities in diagnostic accuracy across different demographic groups. Privacy is another concern, as the use of AI in biosignal processing and medical records must comply with strict data protection regulations to safeguard patient information. Trustworthiness is perhaps the greatest challenge in translating these advances from research to clinical practice, as many DL models function as "black boxes," making it difficult for clinicians to interpret their decision-making processes [123, 129].

While all of these matters should invite to a deeper reflection on the role of AI and its implications, in this work, out of the main societal challenges, we focus on the environmental impact of DL. As we invest in developing ever-more powerful models, what is the cost to our environment, given the substantial energy requirements for training and maintaining these systems?

3.2.1 The environmental impact of Deep Learning

The increasing complexity and scale of AI models have led to significant energy consumption and greenhouse gas emissions. Studies estimate that training large-scale models can be responsible for hundreds of tonnes of CO_2 emissions, which is considerable when compared to the 2 tCO_2eq/person/year limit suggested to keep global warming below 1.5° [23]. The environmental impact of DL primarily result from hardware production, model training, cooling systems, and inference.

In addition to energy consumption, AI models also have a water footprint, which refers to the fresh water used in hardware manufacturing, server cooling, and electricity generation. The production of semiconductors, essential components of AI hardware, requires large quantities of ultra-pure water. Furthermore, data center cooling systems generate wastewater that, if improperly treated, may lead to water contamination and ecosystem damage [50]. The overall water consumption of AI models varies depending on factors such as model size, data center location, energy sources, and cooling system efficiency. The cooling of data centers also generates large amounts of wastewater, contaminated with pollutants, which can lead to the contamination of local water supplies if not properly treated. As AI continues to expand, these environmental concerns are expected to become more critical, reinforcing the need for sustainable AI practices.

Beyond training, AI inference also contributes significantly to environmental impact. Large-scale generative models, such as ChatGPT, require extensive computational resources both for training and continuous inference. Since these models are accessible online, they further engage end-user devices, network infrastructure, and data centers, leading to increased energy consumption and exacerbating issues such as metal scarcity [18]. The digital sector in the European Union alone accounts for 9.3% of electricity consumption and over 4% of greenhouse gas emissions [18]. However, many studies focus only on training energy consumption and estimate its carbon footprint based on the electricity mix used, often neglecting the deployment phase and hardware-related emissions. This narrow focus, sometimes referred to as "carbon tunnel vision", overlooks critical factors such as resource extraction and hardware disposal.

AI's environmental footprint consists of both operational and embodied emissions. Operational emissions arise from energy consumption during data processing, training, and inference, while embodied emissions are linked to hardware manufacturing, transportation, and disposal [212]. The rapid growth of AI has led to a superlinear increase in computational demands, further exacerbating environmental concerns.

Accurately measuring the carbon footprint of AI models remains challenging due to complex interactions between model architecture, hardware, and energy consumption. While direct measurement is impractical, several estimation methods have been proposed. Menghani et al. (2023) categorize footprint metrics, which account only for operational emissions, into three main groups [127]:

- Training and inference metrics: Number of floating-point operations (FLOPs), model size (parameters), and RAM consumption.
- Training-specific metrics: Number of epochs to convergence, model latency during backpropagation, and the number of labeled examples required.
- Inference-specific metrics: Model disk size and latency during the forward pass.

Beyond its environmental impact, AI development and deployment also entail high financial costs and challenges associated with deploying large models. Reducing energy consumption without compromising model performance is a key goal of efficient DL research.

The concept of Green AI, introduced by Schwartz (2020), refers to AI research that aims to achieve progress while considering computational cost [168]. Several approaches have been explored to reduce AI's carbon footprint [203]. One approach is the Precision-Energy Trade-off, which relies on optimizing model efficiency by balancing accuracy and energy consumption: smaller neural networks can significantly reduce energy consumption without major accuracy loss, though there is a threshold beyond which energy savings come at the cost of reduced model performance [218]. Likewise, developing lightweight architectures designed for power efficiency can reduce energy consumption [125]. Other

studied approaches include: subsampling data for reducing computational overhead, applying efficient hyperparameter tuning [47] and energy-aware deployment.

Bouza et al compared seven tools for estimating AI energy consumption [23]. Each tool has distinct methodologies, measuring different aspects such as CPU/GPU usage, memory consumption, and idle power. Some tools estimate full machine-level consumption, while others focus on individual AI processes. Overall, the authors were able to gather a few simple recommendations:

1. Reduce model size as possible;
2. Optimize hyperparameters (batch size, epochs) to reduce unnecessary training;
3. Use renewable energy-powered cloud providers (if available);
4. Enable GPU/CPU power-saving settings;
5. Shut down idle hardware to avoid unnecessary energy drain.

Two other interesting conclusions drawn from the study were that using smaller hardware that fully utilizes its resources is more efficient than using underutilized large clusters and that frequent model checkpointing does not negatively impact energy consumption [23].

Despite the growing recognition of AI's environmental impact, awareness among researchers and practitioners about reducing their carbon footprint remains limited [23]. This raises ethical concerns regarding the justification of large-scale AI pipelines, particularly when marginal improvements in accuracy come at the cost of massive computational resources [193]. As AI continues to advance, greater accountability is needed from researchers, infrastructure providers, and stakeholders to prioritize sustainability in model development [203]. While AI's carbon footprint extends beyond model training to include hardware manufacturing and energy sources, the focus of this thesis is specifically on DL model development. The following section will explore strategies to enhance model efficiency, emphasizing reducing training energy consumption through model simplification and leveraging pre-trained models to minimize unnecessary computations.

3.2.2 Towards more sustainable approaches through model efficiency

In the last few years, attempts have been made to mitigate the footprint of AI and find more efficient solutions. A review paper on energy-efficient DL approaches [124] provides a comprehensive review of strategies for improving energy efficiency in DL across different phases of the lifecycle, namely IT infrastructure, data handling, modeling, training, deployment, and evaluation. The key motivation is the high energy consumption associated with DL models, both in training and inference, leading to significant environmental and financial costs. The study synthesizes previous research to present a holistic view of energy-efficient methods. Beyond reducing the environmental impact of

DL, efficiency-driven approaches also contribute to practical and scalable AI applications, reducing memory usage and computational demands, enabling the deployment of models in resource-constrained environments, including edge devices and mobile applications. Energy-efficient training techniques, such as mixed-precision training and progressive learning, further optimize resource utilization without sacrificing performance. By incorporating these techniques, we move toward a more sustainable paradigm where DL models are not only effective but also accessible, adaptable, and less reliant on energy-intensive infrastructure. Two important aspects for reducing the carbon footprint of DL models are:

- **Reducing model complexity**: smaller models and architectural optimizations lead to less energy consumption
- **Reusing pre-trained models**: leveraging existing models for transfer learning, minimize the need for extensive training.

DL efficiency can be categorized into five key areas, as outlined by [127]: (i) Compression Techniques, which include pruning, which eliminates unnecessary model weights to reduce size and computation time, and quantization, which lowers numerical precision (e.g., from 32-bit floating point to 8-bit integers) to decrease memory usage and increase inference speed (ii) Learning Techniques, consisting of training methods like self-supervised learning, data augmentation, and knowledge distillation, where a smaller model (student) is trained using a larger pre-trained model (teacher) to retain performance while reducing complexity [65]; (iii) Automation, which involves strategies such as hyperparameter optimization and neural architecture search; (iv) Efficient Architectures, focusing on the design of structurally optimized networks; (v) Infrastructure and Hardware, leveraging specialized hardware and software tools for efficiency.

Efficiency in DL refers to achieving the same level of performance while using fewer resources. These resources can include energy, computational power, time, data, and money. While energy efficiency is crucial due to the environmental impact of AI models, adopting energy-efficient approaches often brings additional benefits, such as improved data efficiency, reduced training time, and lower financial costs [208, 93]. Conversely, efforts to optimize resource usage in these areas typically lead to a lower carbon footprint, making efficiency a key factor in both sustainability and cost-effectiveness.

In [68], the authors present a data-efficient DL approach for automated diagnosis from echocardiography using contrastive Self-Supervised Learning (SSL). To address the need for large labeled datasets, which are costly and time-consuming to obtain due to the need for expert annotations, the authors propose EchoCLR, a self-supervised learning method specifically designed for echocardiogram videos. This approach enables the model to learn rich representations from unlabeled echocardiograms, which can then be fine-tuned for specific cardiac disease classifications using only a small amount of labeled data. The efficiency of this method is demonstrated in two key aspects. First, label

efficiency: By leveraging SSL, EchoCLR significantly reduces the dependency on large labeled datasets, showing that only a limited number of annotated examples are needed to achieve high performance. Second, computational efficiency: Because the model learns general representations in an unsupervised manner before fine-tuning, it requires fewer training iterations when applied to a specific classification task. This results in faster convergence and lower overall training costs. The approach employed here is a form of Transfer Learning (TL), a technique that allows models to adapt previously acquired knowledge to new tasks.

3.2.3 What is Transfer Learning?

Deep Transfer Learning, commonly referred to as Transfer Learning, is a subfield of DL that aims to mitigate two major limitations of DL models: the need for large labeled datasets and high computational costs associated with training from scratch [78]. TL enables knowledge transfer from a source domain to a target domain, improving learning efficiency and generalization. This can be achieved by reusing pre-trained models, extracting relevant features, or selecting and re-weighting source data instances to enhance training on the target task or dataset.

TL methods can be categorized based on labeling aspects and applied approaches. Regarding the first categorization, TL can be (i) transductive, when only the source data is labeled, (ii) inductive, if both source and target data are labeled, or (iii) unsupervised, if none are labeled. Regarding the applied approaches, TL can be categorized into four groups [227, 194]:

- **Instance-based**: selecting relevant data samples from the source dataset and adapting their weighting to make the training more adjusted to the target dataset.
- **Feature-based**: mapping instances or features from both source and target data into a common representation space where the differences between domains are minimized, improving feature alignment.
- **Model-based** or **parameter-based**: leveraging knowledge from a pre-trained model on a source task to improve learning on a new target task or dataset.
- **Relational-based**: extracting transferable knowledge from both source and target data by leveraging logical relationships or structural dependencies, such as graphs, rules, or adversarial learning techniques.

Except for model-based and adversarial-based approaches, all other categories have been explored for different ML techniques. Although all these approaches are applicable to DL, DTL is more commonly focused on model-based approaches as these can tackle the domain adaptation directly by re-adjusting the network [78]. As such, this category is the one we will focus on in this work.

TL employs various techniques, including pre-training, freezing, fine-tuning, and adding new layers, to adapt a DL model trained on source data for a new target task. A pre-trained model consists of layers whose parameters have been learned from source data. Freezing layers means keeping their weights constant, while fine-tuning involves initializing some or all layers with pre-trained weights rather than random values, allowing them to be further adjusted for the target data. Progressive Learning is an approach that builds on previously learned tasks by freezing an entire pre-trained model and adding new trainable layers to adapt to the new task, effectively mitigating catastrophic forgetting (which happens when the model, when fine-tuning, loses previously learned information from the source task) [78].

In DL architectures, earlier layers capture low-level features, middle layers abstract patterns, and later layers perform classification or prediction. Given this hierarchical structure, an efficient strategy for TL is to freeze the earlier and/or middle layers of a pre-trained model while fine-tuning the later layers or, in the case of progressive learning, adding new layers to the final portion of the network to enhance adaptability. Progressive Learning is successful in the case of a task transfer for the related source and target data. It cannot deal with the distant source and target data since the earlier layers are frozen and cannot learn new features. The most common TL method is using a trained model on a highly related dataset to target data and finetune it on target data (finetuning) [78].

Despite its strong theoretical foundation and proven success in many applications, TL can lead to performance degradation - known as Negative Transfer - due to several challenges:

- **Catastrophic Forgetting:** When fine-tuning, the model may lose previously learned information from the source task;
- **Bias in Pre-trained Models:** Pre-trained models may introduce biases, leading to poor generalization in the target domain;
- **High Domain Shift:** Mismatches between the source and target distributions can negatively impact performance.

In [139], the authors investigate, under the scope of images, the underlying mechanisms that make TL effective. The aim is to understand which parts of a deep neural network are responsible for successful knowledge transfer from a source domain to a target domain with limited data. The concepts of (high level) feature reuse and low-level data statistics are explored, providing new insights into how pre-trained weights influence model performance. The study demonstrates that successful TL relies not only on reusing high-level features but also on the transfer of low-level data statistics that are preserved even when image features are disrupted. Experiments using block-shuffled images (where visual features are scrambled) show that models benefit from the statistical properties of data, even when traditional feature reuse is hindered. Models fine-tuned from pre-trained weights remain within the same basin of the loss landscape, meaning their optimization

trajectories are smoother and more stable. On the other hand, models trained from scratch on the target domain show greater variability in their learned parameters and often fall into different loss basins. Staying in the same loss basin, however, can be advantageous for similar tasks but detrimental for dissimilar tasks. The challenge in TL lies in balancing the retention of useful knowledge while allowing sufficient flexibility for adaptation to the new task. Fine-tuned models from pre-trained weights exhibit greater similarity in both feature and parameter space, making similar mistakes and converging to comparable solutions. Randomly initialized models, on the other hand, show more diversity in learned features and greater distance in parameter space. Early network layers primarily capture general features that transfer well across domains, while deeper layers are more specialized and sensitive to parameter changes. The study also finds that fine-tuning from earlier checkpoints of pre-trained models can achieve similar accuracy to later checkpoints, offering potential efficiency benefits.

Despite the widespread success of TL in computer vision, its application to time series classification remains relatively underexplored. TL effectiveness when dealing with time series has shown mixed results - some transfers lead to significant accuracy improvements, while others result in negative transfer, degrading performance [46]. A key factor influencing TL success in time series classification is dataset similarity, as transferring knowledge between structurally similar datasets tends to yield better results. In this context, fine-tuning proves to be more effective than freezing layers when adapting 1D CNN models, and selecting source datasets that are not extremely distant helps mitigate negative transfer effects. However, when the distribution gap between source and target datasets is too large, training from scratch can still outperform TL. While TL holds promise for time series classification, its benefits are highly dependent on a careful selection of the source dataset to ensure meaningful knowledge transfer [46].

In [20], the authors trained a multivariate physiological network to classify biosignal type and then applied a TL approach (transferred the learned features to a support vector machine (SVM) classifier) to classify the type of device that was used to record the signal. The study shows that features learned from biosignal classification can be transferred to another classification task; however, TL performance varies - some TL models lead to improvements, but, in certain cases, training from scratch gave better results (negative transfer). The authors' conclusions align with those from the previous study, arguing that the effectiveness of TL in biosignals depends on the similarity between the source and target datasets, the alignment of learned features with the target task, the fine-tuning strategy, and the size of the target dataset. While TL improved classification accuracy in some cases, in others, training from scratch performed better due to insufficient feature transferability or large domain shifts.

3.3 Deep Learning applied to Biosignals

3.3.1 Transforming Biosignal Processing with DL

In Chapter 2, several inherent characteristics of biosignals were explored, including inter- and intra-subject variability, non-stationarity, and susceptibility to noise and artifacts. These factors introduce a level of complexity that makes this type of time-series data particularly challenging to process, analyze, and interpret [122]. Consequently, traditional analysis methods require extensive domain expertise and manual feature extraction, making them both time-consuming and prone to inconsistencies. DL approaches are transforming the field of biosignal processing by automating feature extraction, improving classification accuracy, and enabling real-time analysis. For instance, unlike conventional ML, DL systems do not rely on extensive pre- and post-processing steps [170] nor manual feature engineering and can identify subtle patterns in biosignals that may not be easily detectable by human experts, by leveraging capabilities of each architecture.

RNNs, particularly LSTM and GRU architectures, have been widely employed in physiological signal processing due to their ability to model long-term dependencies and capture temporal patterns in non-stationary data. Bidirectional Long Short Term Memorys (BiLSTMs), which process input sequences in both forward and backward directions, have demonstrated strong performance across various biosignal applications, such as respiratory rate prediction from EMG, Photoplethysmography (PPG), and ECG signals [96]. In this study, BiLSTM outperformed other approaches, but the authors also explored hybrid models that integrate LSTM layers with 1D convolutional layers. Such combinations leverage the strengths of each architecture: CNNs are particularly effective at extracting local spatial and frequency-domain features, while RNNs are well-suited for learning temporal dependencies. This synergy has been successfully applied to biosignal processing tasks, including ECG waveform delineation, segmentation, and classification [108] and stroke prediction [196]. Additionally, both CNNs and RNNs have proven useful in handling noisy biosignal data, with studies demonstrating their effectiveness in noise removal techniques, particularly for ECG signals [84]. The topic of noise handling will be explored further in Chapter 5.

Inter- and intra-subject variability remains one of the most significant challenges in biosignal analysis, even with DL. In theory, if a dataset is sufficiently diverse and representative, a model can learn subject- and context-independent features. However, in practice, acquiring large-scale datasets that encompass multiple subjects and environmental conditions is often impractical. DL strategies to address this challenge align with the generalization techniques discussed in Section 3.1.5. One effective approach is domain adaptation, which has been successfully applied to wearable biosignal data for stress assessment in construction workers, achieving strong generalization across different individuals and contexts [100]. TL strategies play a crucial role in improving the robustness of biosignal processing models when data is scarce, allowing pre-trained models to be

fine-tuned for new applications, making them more applicable to real-world healthcare and occupational settings [20]. To address the challenge of collecting annotated biomedical datasets, a promising strategy is the integration of human-in-the-loop (HITL) methods, where DL models assist human experts in streamlining event annotation [169]. These iterative annotation frameworks significantly reduce manual workload while ensuring high-quality labeled data. Another key approach is SSL, which capitalizes on the vast amounts of unlabeled biosignal data generated by modern wearable devices [150]. SSL pre-trains models using pretext tasks that extract meaningful representations from unlabeled data before applying TL to fine-tune the model with smaller labeled datasets. This technique enhances model generalization and minimizes reliance on extensive manual annotation, making it a powerful tool for improving biosignal analysis in diverse real-world settings.

3.3.2 Relevance in Health Applications

The integration of DL into healthcare has significantly enhanced patient analysis, diagnosis, and treatment by automatically identifying patterns in complex biomedical data. Deep neural networks have proven highly effective in medical diagnosis and patient monitoring through tasks such as biosignal classification and anomaly detection. By automating these processes, DL not only improves diagnostic accuracy but also facilitates personalized treatment strategies tailored to individual patient needs. TL further enhances the applicability of DL models in healthcare, allowing pre-trained models to be adapted to biosignal and medical imaging tasks with minimal labeled data. This capability is especially valuable in clinical settings where annotated datasets are scarce and often very imbalanced, making it a powerful tool for real-world healthcare applications. Despite these advantages, challenges remain in deploying DL models in clinical practice. Issues such as model interpretability, biases in training data, and the need for regulatory compliance are still not fully addressed, hindering the widespread adoption of these technologies. However, ongoing research continues to refine DL systems, ensuring their reliability, fairness, and robustness in medical applications. The continued development of DL for biosignal analysis is essential for addressing challenges such as data heterogeneity, label scarcity, and real-time processing requirements, ultimately improving patient outcomes and medical decision-making [101].

DL has already demonstrated its transformative potential across various healthcare applications, enhancing disease diagnosis, long-term monitoring, and personalized treatment. One of its most impactful applications is in disease detection, where it has been successfully used for diagnosing cardiovascular conditions through ECG analysis [213], identifying sleep apnea [90], and detecting epileptic seizures [6, 59]. In neuromuscular disorders, DL has also been applied to Amyotrophic Lateral Sclerosis (ALS) and Myopathy detection using EMG signals, improving early diagnosis with a lightweight network [152].

Beyond diagnostics, DL plays a crucial role in long-term physiological monitoring

[192, 160], facilitating the continuous tracking of vital signs with applications in wearable healthcare and remote patient monitoring. In the field of mental health, several studies have also been published [162], with stress detection being an extensively explored subject [43, 221], leveraging biosignals such as EEG, ECG, and HRV to assess physiological responses to stressors. Additionally, human activity recognition [126] and sleep tracking [97] are emerging fields where DL enhances the analysis of behavioral and physiological data for improved health assessments.

As AI-driven biosignal processing continues to evolve, DL is poised to revolutionize healthcare by enabling earlier disease detection, continuous monitoring, and more precise therapeutic interventions. However, widespread adoption still requires addressing key challenges such as model interpretability, biases in training data, and regulatory compliance. With ongoing research refining these technologies, DL is expected to play an increasingly vital role in improving clinical decision-making and patient outcomes.

3.3.3 Recent Advances

Transformer architectures represent a recent paradigm in DL and have begun making a significant impact on biosignal processing. As discussed in 3.1.4.3, Transformers leverage self-attention mechanisms to capture long-range dependencies efficiently, while also allowing parallelization, making them particularly suitable for high-dimensional biosignal data. Given their success in NLP, researchers have adapted Transformers for biosignal-related tasks, achieving state-of-the-art performance [10]. In electrocardiography, Transformer-based models have been successfully applied to arrhythmia classification and heartbeat segmentation, outperforming other DL architectures, leveraging their ability to capture temporal dependencies and complex morphological variations. This architecture has also been applied to EEG, namely for brain-computer interface (BCI) applications, emotion recognition, and seizure detection, benefiting from their ability to model intricate neural patterns across time. Similarly, EMG-based applications have used Transformers for silent speech recognition and gesture classification, where self-attention mechanisms enhance robustness to noise and variability across subjects. Beyond unimodal applications, multimodal Transformers have gained traction in integrating biosignals from multiple sources, such as EEG and fNIRS for cognitive state estimation, or EMG and accelerometry for advanced human-machine interfaces [10].

Recent efforts have also focused on improving Transformer efficiency in biosignal applications. Variants such as Temporal Fusion Transformers (TFT) and Time Series Transformers (TST) have been proposed to address the computational challenges of long biosignal sequences while preserving rich temporal dependencies. Additionally, lightweight Transformer architectures optimized for real-time processing have been introduced for wearable devices and mobile health applications, ensuring scalability without compromising performance. These advancements solidify Transformers as a powerful tool in biosignal analysis, with ongoing research exploring ways to enhance their interpretability,

efficiency, and adaptability across diverse biomedical applications.

A recent advance in biosignal processing involves the integration of Large Language Models (LLMs) with biomedical signal analysis, as demonstrated by the development of SignalGPT, a system designed to convert biomedical signals into structured text reports [109]. This approach leverages the power of LLMs to interpret physiological signals such as ECG, EEG, and EMG by transforming them into technical descriptions that can be further analyzed by AI models. The system consists of a biomedical signal processing pipeline and a fine-tuned ChatGPT model, which work together to generate clinical reports, enhancing interpretability and reducing the burden on medical professionals. SignalGPT was validated using ECG data, demonstrating its feasibility in providing accurate signal descriptions and medical insights.

Emerging advances in DL for biosignal processing have increasingly focused on efficiency, transferability, and scalability. Beyond architecture-based improvements, the idea of creating foundation models for biosignals is gaining traction as a promising approach to enhance efficiency and generalization. Instead of training models from scratch for each application, these models leverage large-scale self-supervised learning techniques, allowing for robust feature extraction from vast amounts of unlabeled biosignal data. For instance, a recent study trained foundation models for ECG and PPG using SSL on over 141,000 participants' wearable data to create highly generalizable representations, significantly reducing the reliance on annotated datasets [1]. These pre-trained models can be fine-tuned on smaller, domain-specific datasets using TL, making DL solutions more accessible for real-world healthcare applications.

A key consideration in recent advances in biosignal processing is the need to develop lightweight, efficient models and hardware solutions that are not only computationally effective but also practical for real-world applications.

A very recent and innovative approach named *MOMENT: A Family of Open Time-series Foundation Models* is aimed at improving time-series analysis by creating a suite of pre-trained models that can be applied to various time-series tasks [56]. The key challenge in pre-training such models lies in the lack of a large, cohesive public repository of time-series data and the diverse nature of time-series characteristics, such as varying sampling rates and missing values. To address this, *MOMENT* introduces the Time Series Pile, a diverse collection of publicly available time-series datasets spanning multiple domains. By leveraging masked time-series prediction as a pre-training task, *MOMENT* enables models to handle various time-series applications like forecasting, classification, anomaly detection, and imputation, even with minimal supervision. While this system was not designed specifically for biosignals, it includes ECG signals as one of its time-series examples, suggesting that incorporating other biosignals into the framework is feasible. The models demonstrate strong performance in scenarios where computational resources are limited, providing a versatile foundation for time-series analysis and setting a new standard for evaluation through comprehensive benchmarking against existing methods.

3.4 Final Remarks

DL has revolutionized biosignal processing, offering automated feature extraction, improved performance, and the ability to model complex temporal dependencies. This chapter has explored the foundational concepts behind DL, key architectural paradigms, and their applications in biosignal analysis. While these models have demonstrated remarkable capabilities, they also present significant challenges, including high computational costs and difficulties in generalizing across different domains.

One of the major concerns is the resource-intensive nature of DL models. The environmental footprint of training and deploying these models has raised important ethical and sustainability considerations, driving research into more efficient training methods and lightweight architectures. Transfer learning, and energy-efficient optimization techniques have emerged as viable strategies to mitigate these concerns while maintaining high performance.

Additionally, while DL has proved, from the existing literature, that it is efective in handling the inherent complex characteristics of biosignals, such as their non-stationary and noisy nature, the challenge of generalization remains a key issue in biosignal processing. Variability in biosignals due to subject-specific differences and changing acquisition conditions complicates model deployment in real-world applications. While domain adaptation and transfer learning offer promising solutions, the risk of negative transfer highlights the need for careful dataset selection and fine-tuning strategies.

Despite these challenges, recent advances in DL have shown great promise. Current trends indicate a growing emphasis on unifying biosignal processing frameworks instead of developing isolated, task-specific solutions. Recent efforts introduced generalizable pipelines that integrate pre-trained models with domain adaptation and multi-task learning strategies [56, 1]. This aligns with the broader movement towards reusable AI models, where instead of reinventing processing methods for each biosignal modality, researchers leverage common architectures that can be adapted to different tasks and datasets. This shift towards standardization, scalability, and efficiency underscores the future of DL in biosignal analysis, paving the way for more efficient, robust, and widely deployable AI-driven healthcare solutions.

4

Deep Learning applied to Cardiovascular Disease Classification from ECG records

In this chapter, we present a study on the application of Deep Learning (DL) methods for CVD classification using ECG records. The study was conducted using ECG data from the PTB-XL dataset, a large publicly available resource. We utilized the data in its raw format as well as in alternative representations, evaluating whether combining different formats could enhance diagnostic reliability.

4.1 Introduction

In a healthy heart under normal conditions, approximately once every second, an action potential is generated at the sinoatrial node and sequentially transmitted to the atria, the atrioventricular node, the bundle of His, the Purkinje fibers, and, finally, to the ventricles. This rythmic process can be captured in an electrocardiogram through the use of at least two electrodes in opposite sides of the heart, which capture the potential difference between those two points. In this case, we refer to it as one-lead ECG; more electrodes can be used, and every combination of two electrodes provides an additional lead, which offers a different perspective of the same phenomenon [52]. The most common ECG configurations include 12-, 6-, 3-, and single-lead setups. While the 12-lead ECG is the gold standard in clinical settings due to the comprehensive information it provides, wearables typically use single-lead ECG to allow for a simpler and more portable setup. The visual representation of this process is a periodic signal in which several waveforms are distinguishable, represented in Figure 4.1, corresponding to the different phases of depolarization and repolarization of the heart's structures: a P wave (generated by the atrial depolarization), a QRS complex (generated by the ventricular depolarization), and T and U waves (generated by the ventricular repolarization). The shape of these waveforms and the rhythm (or lack thereof) at which they are generated are informative regarding the presence of cardiac abnormalities, which is why these exams are commonly performed in

clinical contexts [52, 104].

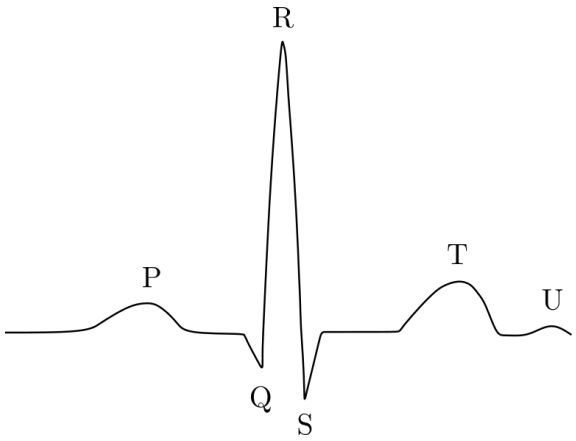


Figure 4.1: Illustration of the ECG waveforms.

cardiovascular disease (CVD) is a collective term used for all disorders that affect the blood circulatory system and, according to the World Health Organization (WHO), these represent the number one cause of death worldwide [28]. Electrocardiography can play a valuable role in the early identification of CVD either directly through the recognition of abnormal waveforms (in diseases that affect the heart's electrical conduction) or by detecting irregularities in the heart rate, especially when performing continuous monitoring with wearable sensors. Additionally, changes in the heart rate or its variability can be informative even in the absence of diseases, making ECG a useful tool for prevention [195]. However, one of the biggest challenges in expanding the use of ECG is the requirement for effective data processing tools - an area where DL offers significant potential by automating complex processing steps.

Since manual ECG analysis is an arduous and challenging task, from early ages, efforts have been made to create automated methods for supporting and reducing clinicians' workload. Several have proved to be efficacious while also offering faster and less subjective results [61, 3, 112]. DL, as discussed in Chapter 3, facilitates ECG analysis by enabling the direct use of raw signals and automating processing steps that would otherwise require some level of expertise to be carried out. Specifically, in the context of ECG signal analysis, DL has, for instance, been successfully applied to: R-peak detection [99], heart disease classification [111], identification of arrhythmia [215], atrial fibrillation [161], congestive heart failure [206], and in heartbeat classification in continuous monitoring [160].

While most literature on the application of DL methods to ECG relies on the 1D signal, recent studies have explored two-dimensional (2D) representations of the signal, yielding improved results [112, 3]. Such findings suggest that varying formats capture distinct information or, at least, highlight different aspects of the same data, ultimately influencing the outcomes. Moreover, each format entails specific architectural choices with varying levels of complexity. From this reasoning, some questions emerge. If diverse representations indeed convey different information, could their combined use enhance

performance? Is the increased complexity justified by potential improvements? Can we establish a clear relationship between model complexity and performance in the current context? In this work, we investigated and compared the performance of various DL approaches, spanning a range of complexity, for CVD classification from ECG, utilizing raw 1D signal, 2D representations, and multimodal approaches that integrate both.

4.2 The PTB-XL dataset

The PTB-XL dataset is a large publicly available dataset containing 21,837 12-lead ECG records. The 10-second signals were collected from 18,885 different subjects, including both healthy and non-healthy subjects. The records have been labeled by up to two cardiologists and the comorbidities that are present in the data are the following four: Myocardial Infarction (MI), ST/T Changes (STTC), Conduction Disturbance (CD), and Hypertrophy (HYP). The records that do not reflect the presence of any disease are annotated as Normal (NORM) and encompass 56.36% of the dataset. The subjects' age ranges between 0 - 95 years and are evenly distributed in terms of gender [205].

The dataset is stored in a 16-bit binary format with a resolution of 1 μV/LSB. The 12-lead ECG signals were originally acquired at 400 Hz and two versions of the data are provided in PhysioBank: an upsampled version at 500 Hz and a downsampled version at 100 Hz. Although the ECG data is predominantly of high quality, the PTB-XL dataset includes metadata on signal quality. Some instances exhibit static noise (14.9%), baseline drift (7.4%), burst noise (2.8%), or electrode-related issues (0.1%) [205].

4.3 Cardiovascular Disease Classification: A Comparative Study of 1D and 2D Representations and Multimodal Fusion Approaches[1]

4.3.1 Objectives

Several DL-based methods have been developed for ECG classification, as reported in section 3.3. Most of these approaches for ECG analysis use the raw 1-dimensional (1D) ECG signal. However, recent studies have explored 2-dimensional (2D) representations by converting the ECG signal into images, with some demonstrating that models leveraging these representations can achieve superior performance compared to those using the traditional 1-dimensional (1D) signal [112, 3].

Distinct data representations offer distinct information levels, therefore, contributing differently to classification performance. As such, classification tasks, namely in the

[1]This section reports joint work with Hemaxi Narotamo (2D models) and Ricardo Santos (Fusion Models) and it is based on the publication "Deep Learning for ECG Classification: A Comparative Study of 1D and 2D Representations and Multimodal Fusion Approaches" by Hemaxi Narotamo†, **Mariana Dias**†, Ricardo Santos†, André V. Carreiro, Hugo Gamboa, and Margarida Silveira, published in the journal Biomedical Signal Processing and Control (2024). † First authors.

context of disease diagnosis, may benefit from complementary information from distinct modalities [15]. In this work, our main goal was to perform CVD classification on the PTB-XL dataset. For that, we firstly explored different 1D approaches, testing not only GRU- and LSTM-based networks but also combinations with 1D-CNN layers, as well as with Attention mechanisms. Additionally, we tested the potential of transforming the ECG signals to 2D images and process them with 2D-CNN-based models. These transformations were proposed by Ahmad et al. [7] and are illustrated in Figure 4.4: Recurrence Plot (RP), Markov Transition Field (MTF) and Gramian Angular Field (GAF). Finally, we analised the value of performing multimodal fusion in this context, combining 1D signal and 2D image transformations, offering new insights into the effectiveness of hybrid models in medical signal processing.

4.3.2 Related Work

4.3.2.1 1D ECG Classification

Several studies have utilized the PTB-XL dataset to train DL models for CVD classification using 1D ECG records. Namely, in [181] and [182], the authors performed CVD classification from raw ECG signals and features extracted from it using a combination of 1D CNN and fully-connected layers. Strodthoff et al. in [188] compared different models (such as 1D CNN, LSTM, and feature-based algorithms), having shown that architectures based on ResNet50, InceptionTime and bidirectional LSTM exceed feature-based models.

Virgeniya et al. [204] proposed a GRU with an Extreme Learning Machine model for ECG classification of the four diseases present in PTB-XL. Their results were enhanced when the authors applied a process to handle the class imbalance that exists in the dataset, using the Adaptive Synthetic (ADASYN) data sampling process. With this class imbalance handling approach, the authors were able to improve their results from a mean specificity and sensitivity of 58.8% and 92.8% to 91.0% and 86.8%, respectively.

In [214] the authors propose a multi-scale network, based on the ResNet architecture, containing attention modules to combine different viewpoints, i.e. signals from different ECG leads. They obtained an overall AUC and sensitivity of 93.25% and 81.29%, respectively.

4.3.2.2 2D ECG Classification

As mentioned in Chapter 3, 2D-CNNs are a type of artificial neural network used for image segmentation, recognition, classification and processing. In the area of ECG classification for CVD diagnosis, researchers started using approaches that transform the raw ECG signal into 2D images that are then fed to CNNs to perform classification [112, 7, 183].

Most of these methods convert the 1D ECG signal into a 2D time-frequency spectrogram using Short Time Fourier Transform (STFT) [71, 41, 199]. This allows the inclusion of time and frequency details into a single 2D image. For instance, Huang et al. [71] proposed a

method to classify 5 types of arrhythmias from the ECG using a 2D CNN. The proposed approach was compared with a 1D CNN that takes as input the 1D ECG signal. The results showed that the 2D CNN obtained an accuracy of 99.00%, outperforming significantly the 1D CNN which achieved an accuracy of 90.93%. Moreover, they have shown that manual pre-processing steps (signal filtering, feature extraction and selection) are not needed when classifying ECG signals using 2D CNNs. Diker et al. [41] and Ullah et al. [199] also converted the ECG signals into spectrogram images as done in [71]. On the one hand, in [41] these images were used as input for different CNNs: VGG-16, ResNet-18 and AlexNet. The best results were obtained with the AlexNet model. On the other hand, in [199] a 2D CNN was trained to classify ECG images into 8 distinct arrhythmia classes. Although in [41] and [199] it is shown that 2D CNNs can be used for accurate ECG classification, in these works the 2D CNNs performance is not compared to approaches using the 1D ECG signal.

Recently, Ahmad et al. [7] proposed three statistical methods to convert the ECG signal into images conserving the temporal details of the 1D signal: Recurrence Plot (RP), Markov Transition Field (MTF), and Gramian Angular Field (GAF). They used two datasets that contain classification labels for each heartbeat. Thus, they converted each 1D heartbeat into three images using RP, MTF, and GAF, and performed classification using the AlexNet model. The authors concluded that the proposed ECG-to-image conversion approaches outperform previously proposed methods based on time-frequency analysis (such as the STFT).

Therefore, in this work, we aim to apply the image transformations proposed in [7] to ECG records from a different dataset and compare the performance of state-of-the-art CNNs designed for image classification. Since in [7] a dataset that contains one heartbeat per record was used, the classification is performed based only on single-beat information. Herein, we will use not only the single-beat but also beat-to-beat information. Thus, we will convert the ECG signal of 10 seconds into images instead of converting the ECG signal corresponding to only one heartbeat. This will allow us to include more information to perform classification and potentially capture non-continuous changes among multiple heartbeats.

4.3.2.3 Multimodal Fusion

Simple phenomena might be explained through multiple factors. For instance, diseases occur after pathological processes, which can be evaluated by clinical exams, each measuring a specific parameter. Often, multiple tests are needed before making a diagnosis. Therefore, ML models should benefit from all available data. Multimodal Artificial Intelligence (AI) refers to all techniques used to combine individual modalities, which refer "to the way in which something happens or is experienced" [15].

According to Baltrusaitis et al., [15], the multimodal AI topic studies five different challenges: representation, translation, alignment, fusion and co-learning. For this work,

the representation and fusion problems take more relevance. Representation refers to the problem of learning the necessary transformations that must be applied to each modality, so that not only they can be interpreted by a model, but also leverage the optimal information from each source. On the other hand, fusion refers to the methods applied to merge the information of all modalities in the prediction of a given task. Recent work on DL approaches has led to an intersection between these two challenges, as such models combine the representation learning in an initial layer, to the classification performed on the last ones.

Regarding fusion, Huang et al. [72] divide fusion strategies to deal with multimodal data into three types: early, late or joint fusion. Early fusion refers to all methods that aggregate different modalities at the feature level, either original sensory features, manually estimated or taken from a separate network. Then, a single model is trained for the task. On the other hand, late fusion regards the combination of output probabilities from single-modality models. It includes shallow combination methods, such as averaging or major voting, or other models which use individual predictions as input. The joint or intermediate fusion leverages the potential of neural networks for iterative optimization. These methods take features extracted from either earlier or later layers of individual networks, which are then merged into a single pipeline where the prediction occurs. The difference for early fusion is that the propagated final loss affects the unimodal networks, providing a way for parameter adaptation. These can be randomly initiated (end-to-end training) or taken from the previous individual training (fine-tuning).

In [7], the authors resort to RP, MTF, and GAF image transformations, to predict arrhythmia and Myocardial Infarction (MI) with two different strategies. Firstly, an early-like fusion, or multimodal image fusion (MIF), is designed to concatenate images in-depth and perform the classification through a single model. The second is a multimodal feature fusion (MFF) approach, where the first three AlexNets are trained to extract features from the RP, MTF, and GAF images separately. Thereafter, these features are concatenated using a gated fusion network and fed to a Support Vector Machine (SVM) to perform the classification. More recently, in [64], a new DL model called Multimodal Multi-Instance Learning Neural Network (MAMIL) is proposed for an ECG classification task. MAMIL uses as multimodal inputs the original ECG signals and GAF images and it applies multi-instance learning (where each ECG signal and GAF image are treated as instances and each heartbeat is treated as instances), and a novel attention mechanism-based feature fusion method for effective classification. One major drawback of this approach is its high computational consumption.

On other topics, Rastgoo et al. [155] presented a driver stress level classification system using ECG, vehicle and contextual data. Individual CNNs were designed to extract features, whose outputs were concatenated and sent to an LSTM to perform classifications. The training was performed end-to-end and the authors state that results outperform models that leverage traditional features. Hammad et al. [63] proposed a biometric authentication system using ECG data and fingerprints, by comparing a sequential and

parallel system. While the first leveraged modalities separately, the second resorted to individual CNNs to perform feature extraction and a Q-Gaussian SVM to produce predictions. The multimodal results exceeded those from the unimodal approaches, however, limited to differences of around 1%.

Following different fusion strategies proposed in the literature [72], we will explore how multi-label ECG classification can improve from multiple data representations, namely the original 1D signal and the RP, MTF, and GAF image transformations. Therefore, early, late and joint fusion multimodal strategies will be used leveraging developments of unimodal approaches.

4.3.3 Methodology

In this work, we proposed to evaluate how different ECG signal representations perform in a multi-label CVD classification task. For this purpose, two different signal representations are used in different models. Firstly, Recurrent Neural Network (RNN) leverage the temporal dependency between values in the 1D signal; several different 1D architectures are tested and compared (1D CNN-, LSTM-, and GRU-based) regarding their performance on a multi-label classification task on the PTB-XL dataset, previously described in Section 4.2. Then, three signal-to-image transformations, proposed in [7], are applied, and the classification performance is explored using Convolutional Neural Networks (CNN). Furthermore, we resorted to multimodal fusion techniques as a way to leverage the multiple representations of the ECG signal. The best-performing 1D and 2D unimodal networks are merged into a single network resorting to different fusion strategies.

The objective is focused on the classification of each record in any of the four diseases, acknowledging that a record may be associated with multiple labels. The absence of any of these four diseases is implicitly classified as a Normal ECG (NORM). This strategy was adopted to let models distinguish characteristics and patterns of each disease, under the assumption that a Normal ECG would be indicated by negative predictions for all four disease classes. However, to comprehensively assess the models' effectiveness, performance metrics for the Normal ECG/Absence of Disease (AD) class were also calculated.

4.3.3.1 Data Pre-processing

In PTB-XL, the ECG data is available in two sampling frequencies: 100 and 500 Hz. As a frequency of 90 Hz has been described as the minimum sampling frequency suitable for detecting the distinct waveforms in the signal (P, QRS, and T) [24], we opted to use the 100 Hz data.

A second-order Butterworth bandpass filter was applied to the ECG signals, with a high pass cut-off frequency of 1 Hz for baseline wander suppression and low pass cut-off frequency of 45 Hz for high-frequency noise elimination [105], [153], [180].

Table 4.1: Description of records per label in the PTB-XL dataset, with the partition into the train, validation, and test sets, as suggested in [205], with the number of samples in each multi-label disease class.

Acronym	Superclass	No. Records			
		Train Set	Validation Set	Test Set	Total
AD	Normal ECG	7 254	916	913	9 083 (42.4%)
MI	Myocardial Infarction	4 389	544	553	5 486 (25.6%)
STTC	ST/T change	4 193	534	523	5 250 (24.5%)
CD	Conduction Disturbance	3 912	497	498	4 907 (22.9%)
HYP	Hypertrophy	2 121	271	263	2 655 (12.4%)
	Total Samples	17 111	2 156	2 163	21 430

Since the use of all 12-lead signals would imply great computational expense, we decided to use only three leads: I, II and V2. This decision was made based on the fact that these three leads have been successfully used to reconstruct the 12-lead ECG in [226] and, as such, contain most of the relevant information from all channels.

We considered the training/validation/test split recommended by Wagner et al. [205], resorting to the initially provided data partitions. The authors [205] suggested a 10-fold train-test split, which was obtained using stratified sampling and ensuring that all records for a certain patient were placed in the same fold to avoid any leakage. The records in folds 9 and 10 have undergone at least one human evaluation thus they have good label quality. As a result, we followed the advice and used folds 1-8 as the training set, with folds 9 and 10 serving as the validation and test sets, respectively. Furthermore, we excluded samples without ground truth labels. The final number of training, validation and test samples is 17 111, 2 156 and 2 163, respectively (Table 4.1).

In the following subsections, we describe the details of the compared approaches for ECG classification. In all strategies, there are four output nodes corresponding to the four diseases in the dataset (MI, STTC, acrshortCD, and HYP). The absence of any of these diseases corresponds to the "NORM" class of the dataset. Since we are dealing with a multi-label classification problem, we apply a sigmoid activation function to each node in the output layer. To mitigate the problems raised by the dataset imbalance, the models are trained to minimize the weighted binary cross-entropy loss:

$$Loss = -\frac{1}{N}\sum_{o=1}^{N}\sum_{c=1}^{M} w_c[y_{o,c}\log(p_{o,c}) + (1-y_{o,c})\log(1-p_{o,c})] \tag{4.1}$$

where $M = 4$ is the number of diseases, y is a binary indicator of the ground truth class label for observation o, and p is the predicted probability of observation o belonging to class c, N is the total number of observations, and w_c is the weight for class c. The weight for each class is inversely proportional to the number of samples of that class in the train set.

For each approach, the model's parameters used for evaluation on the test set are saved from the epoch where the lowest validation loss was achieved during training.

4.3.3.2 1D Approach

The 1D network had as input the ECG signals after pre-processing. As such, the input consists of three sequences of length 1 000 (each sequence contains 10 seconds recorded at 100 Hz). These input signals were used to train models with different architectures. We explored GRU and LSTM networks (both unidirectional and bidirectional) and the combination of 1D CNN with RNN (GRU/LSTM) layers. Additionally, we explored the use of a self-attention mechanism in the best-performing network. The different applied approaches are denoted as GRU, BiGRU, LSTM, BiLSTM, 1DCNN+GRU, 1DCNN+LSTM, and GRUAtt. The generic structure of these networks is illustrated in Figures 4.2 and 4.3. The bidirectional models (BiGRU and BiLSTM) consist of GRU/LSTM layers that process the time series from both the forward and backward directions.

Regarding the RNN models (LSTM and GRU models), the number of hidden layers $\{2, 3\}$ and the number of hidden units $\{128, 256\}$ were optimized, as well as the dropout rate $\{0, 0.3, 0.5\}$. Dropout was applied to all RNN hidden layers' outputs except the last one, only in non-recurrent connections, as applying dropout to recurrent connections would break the information flow through time. The last state of the last RNN layer was used as input to the fully connected layer, except in the cases of BiGRU and BiLSTM, where the input was a concatenation of the last timestep from the "left-to-right" direction and the first timestep from the "right-to-left" direction.

For the CNN-RNN models, the 1DCNN configuration was composed of three 1D-convolutional layers, each followed by a max pooling and a dropout layer. Some experiments were performed to optimize the number of hidden units $\{128, 256\}$ of the single RNN layer and the dropout rate $\{0, 0.3\}$ (applied to the 1D-convolutional layers).

All models were trained using the *Adam* optimizer, for 200 epochs with a batch size of 256 (or 128 when the allocated memory exceeded the available limits) and an initial learning rate of 0.005.

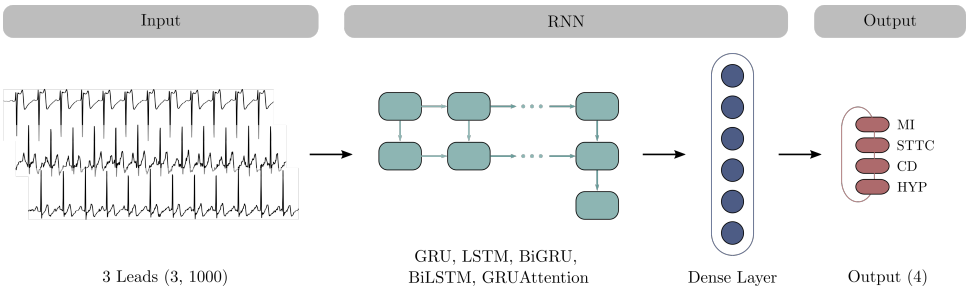


Figure 4.2: Overview of the proposed RNN-based framework for ECG classification.

4.3.3.3 2D Approach

In image classification, the most commonly used CNNs include the AlexNet, VGGNet and ResNet [26], which have exhibited excellent performance in image classification applications throughout the years. The AlexNet has shown enhanced performance in previous

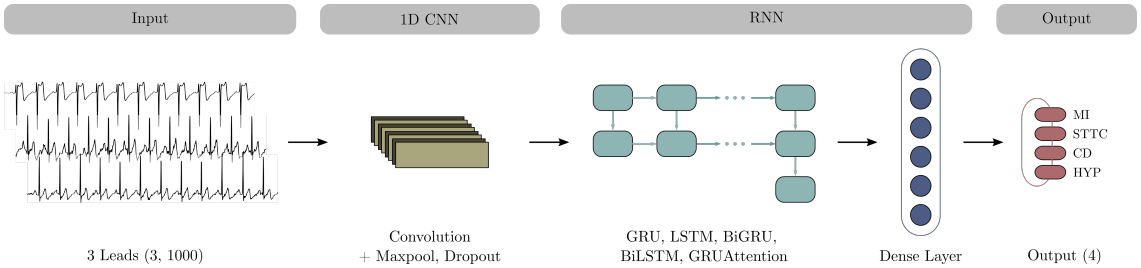


Figure 4.3: Overview of the proposed CNN-RNN based framework for ECG classification. MaxPool denotes the max pooling layer.

works for ECG classification [7, 41]. Thus, we investigated these three architectures, and we also explored more modern approaches, such as MobileNetV2 [163] and AlexNet with a self-attention mechanism (AlexNetAtt) [201], to take advantage of the latest developments in the field.

For each record, we have three signals of size 1 000. Each signal is converted into the RP, MTF, and GAF images [7][2], hence an image of size (3, 1 000, 1 000) is obtained. The ECG to image conversion methods applied to a record based on the lead I signal are shown in Figure 4.4.

Considering a normalized ECG signal with n samples (s_k, for $k = 1, ..., n$ and $0 \leq s_k \leq 1$), the three images (RP, MTF, and GAF) are computed as follows:

- GAF: for each sample s_k calculate the angle $\beta_k = arccos\,(s_k)$, thereafter each pixel (k, l) of the GAF image is computed as $cos(\beta_k + \beta_l)$;
- RP: each pixel (k, l) of the RP image represents the Euclidean distance between two ECG samples (s_k and s_l);
- MTF: firstly define B quantile bins of the signal and assign each sample s_k to the corresponding bin b_j ($j \in [1, B]$), build a $B \times B$ matrix where each entry (i, j), denoted by w_{ij}, represents the probability with which a sample in quantile b_j is followed by a sample in quantile b_i, thereafter each pixel (k, l) in the MTF image represents $w_{ij|x_k \in q_i, x_l \in q_j}$. In this work, B was set to 10, as done in [7].

By concatenating the images obtained for each signal we obtain an image of size (9, 1 000, 1 000). Due to memory constraints, the images of each record were resized to (9, 256, 256). These images and corresponding labels were used to train five CNNs: AlexNet, ResNet50, VGG16, MobileNetV2 and AlexNetAtt. An overview of the proposed image *sub-net* framework for ECG classification is depicted in Figure 4.5.

For the CNNs, we performed several experiments by changing the number of filters in the first layer $\{8, 16, 32, 64\}$, which are then used to compute the number of filters in the following layers, learning rate $\{0.1, 0.01, 0.001\}$, dropout rate $\{0, 0.3, 0.5\}$ and optimizer

[2]The ECG to image transformations are based on a publicly available implementation by Ahmad et al. [7] https://github.com/zaamad/ECG-Heartbeat-Classification-Using-Multimodal-Fusion

$\{Adam, SGD\}$. The models were trained from scratch for 100 epochs with a patience parameter of 20 epochs using SGD.

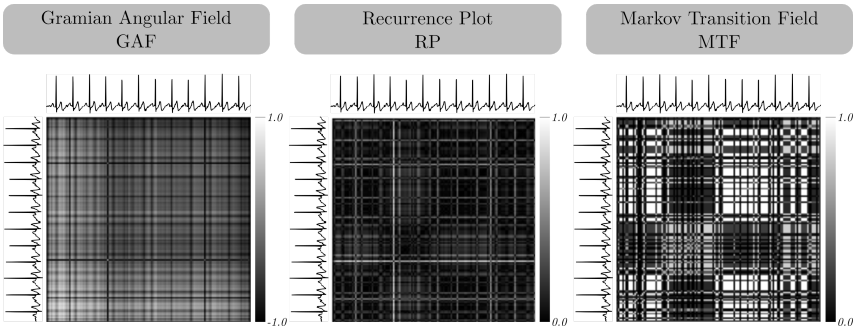


Figure 4.4: Transformation of an ECG signal of 10 seconds (normalized between 0 and 1) into 2D images using RP, MTF, and GAF.

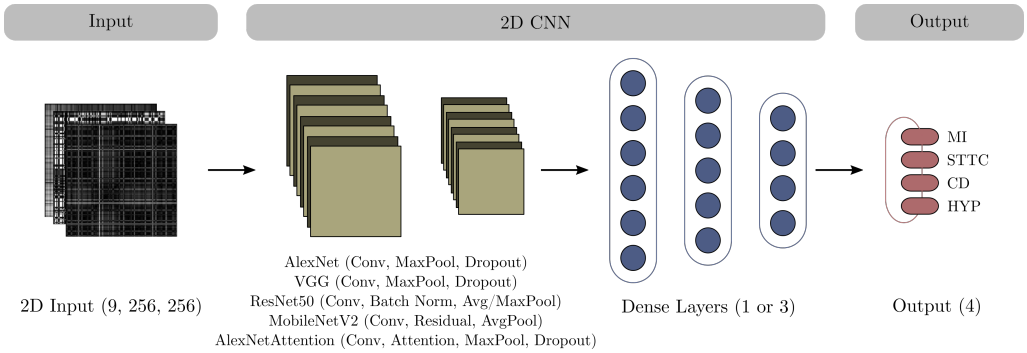


Figure 4.5: Overview of the proposed CNN-based framework for ECG classification. Conv, MaxPool, AvgPool and BatchNorm denote the convolutional, max pooling, average pooling and batch normalization layers, respectively.

4.3.3.4 Multimodal Fusion

In this work, multimodality was leveraged to combine structurally different data types. The raw 1D ECG and its 2D representations are not common modality pairs, such as image and text, and neither are collected through distinct acquisition devices. However, they may provide networks with complementary information, so one could potentially benefit from merging both sources. Therefore, we compared the performance of multimodal networks combining the best-performing single representation networks, namely the GRU network for the 1D signal and the AlexNetAtt for the produced 2D representations. Three different fusion approaches were tested [72], which leverage distinct outputs and training strategies. Additionally, feature-level attention was used in all multimodal fusion strategies, to assert whether such mechanisms aid in the multi-label CVD classification tasks [14, 201]. A self-attention layer was added after unimodal feature concatenation and compared to the original fusion strategies without attention.

Firstly, a simple late fusion approach was developed, leveraging the predictions from individual networks. As depicted in Figure 4.6, a single fully connected layer transforms the

concatenated predictions from each single-modality network, using their static pre-trained parameters. Multiple hidden combination layer sizes were tested, $\{64, 128, 256, 512\}$, as well as multiple learning rates, $\{0.1, 0.01, 0.001\}$. The late fusion batch size was 256. All multimodal networks were trained using a maximum of 200 epochs with an early stopping patience of 10 epochs, no dropout nor L2 regularization, and the *Adam* optimizer. The epoch presenting the lowest validation loss was selected and evaluated on the test set.

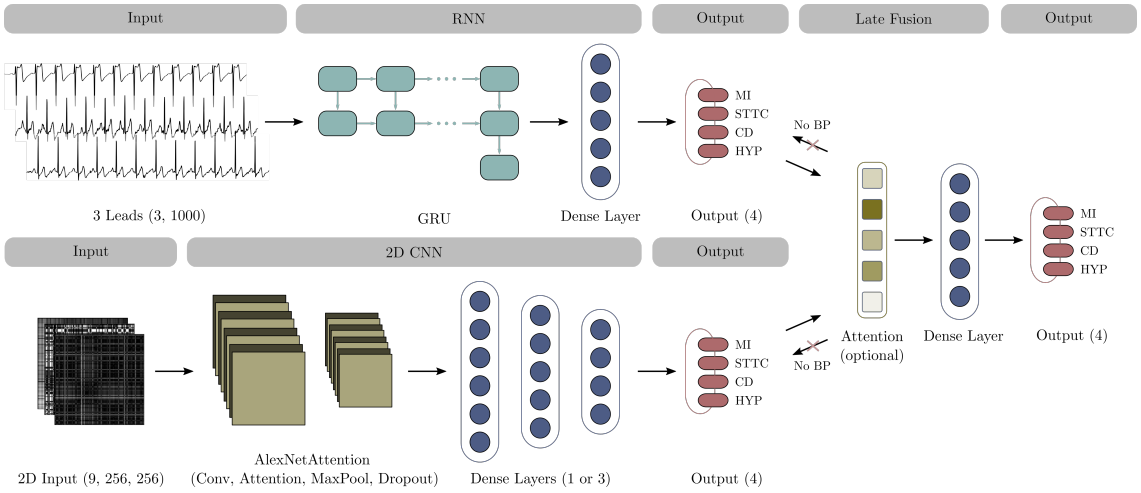


Figure 4.6: Overview of the proposed late fusion framework for ECG classification. Conv denotes the convolutional layer, MaxPool is the max pooling layer, and BP represents the backpropagation to the unimodal networks, which does not occur in the late fusion strategy (red arrow). Additionally, the effect of a self-attention layer after feature concatenation was tested against using solely a single fully-connected layer.

Then, an early fusion network extracted features from both networks, using static pre-trained parameters, as depicted in Figure 4.7. Activation outputs from multiple layers were hooked from unimodal networks. These latent representations become the inputs for the early fusion. While the activation from the GRU block of the 1D network was used in all experiments, activation after the third, fourth and fifth convolution blocks from the AlexNetAtt was tested. To counteract the discrepancy between activations of the 1D and 2D networks, additional max pooling and a fully connected layer reduce the output shape from the later modality. Then, three fully connected layers perform the prediction after the concatenation of both feature sets. As in late fusion, the impact of a self-attention layer after unimodal features concatenation was compared to solely using fully connected layers. Learning rates of $\{0.1, 0.01, 0.001\}$ and final hidden layer sizes from $\{128, 256, 512\}$ were tested during hyperparameter optimization. Each prior fully connected layer doubled the number of neurons. The early fusion training used a batch size of 128.

Lastly, a joint fusion approach was developed, to evaluate the benefit of fine-tuning pre-trained parameters from individual networks towards leveraging complementary information. As schematized in Figure 4.8, activations from the last convolution block or after the first and second fully connected layers were retrieved from the 2D branch, while the activation from the GRU block was always used from the 1D network. Again,

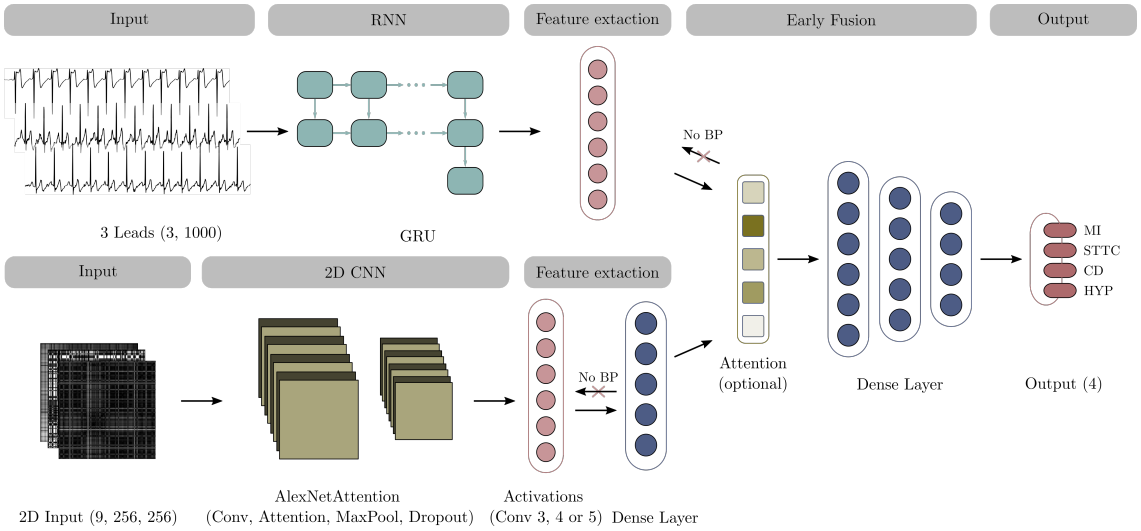


Figure 4.7: Overview of the proposed early fusion framework for ECG classification. Conv denotes the convolutional layer, MaxPool is the max pooling layer, and BP represents the backpropagation to the unimodal networks, which is blocked in early fusion (red arrow). Additionally, the effect of a self-attention layer after feature concatenation was tested against using solely the three fully-connected layers scheme.

an additional fully connected layer was included for dimensionality reduction after the 2D network, to match both unimodal activation sizes. Then three fully connected layers are trained from scratch to optimize the CVD prediction. The performance of such a multimodal fusion network was compared to one using an additional self-attention layer after unimodal output concatenation. The optimization of the joint fusion strategy asserted performances using a batch size of 64, and the tested final layer sizes were $\{128, 256, 512\}$, including a double factor to each previous layer.

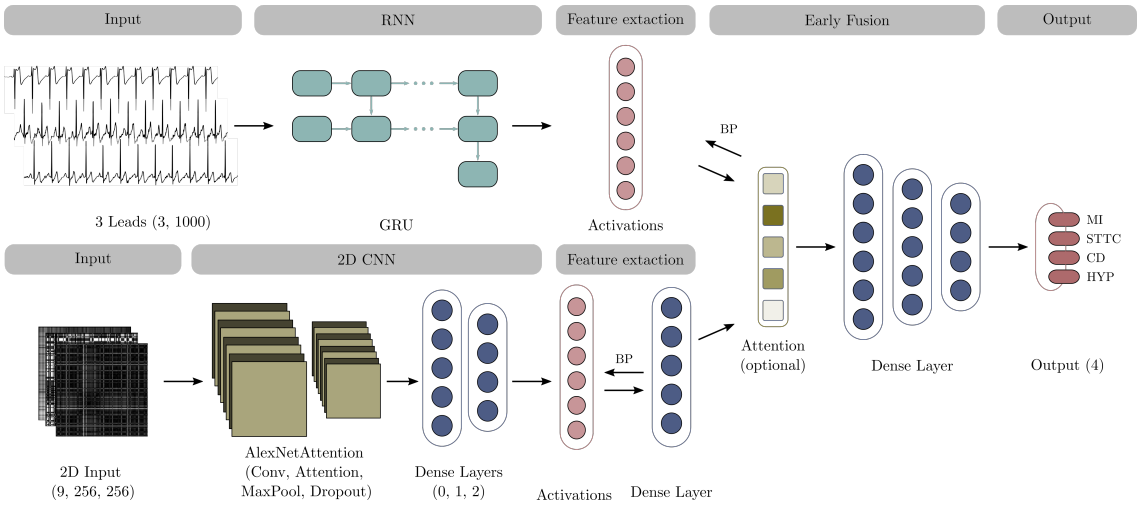


Figure 4.8: Overview of the proposed joint fusion framework for ECG classification. Conv denotes the convolutional layer, MaxPool is the max pooling layer, and BP represents the backpropagation to the unimodal networks, which occurs in the joint fusion setting (green arrow). Additionally, the effect of a self-attention layer after feature concatenation was tested against using solely the three fully-connected layers scheme.

4.3.3.5 Evaluation Metrics and Threshold Optimization

CVD diagnosis from the PTB-XL dataset consists of a multi-label classification problem. Therefore, metrics are computed for each disease label, namely the sensitivity (Eq. 4.2) and the specificity (Eq. 4.3):

$$\text{Sens} = \frac{TP}{TP + FN} \tag{4.2}$$

$$\text{Spec} = \frac{TN}{TN + FP}. \tag{4.3}$$

TP represents true positive predictions, FP indicates false positives, TN stands for true negatives, and FN denotes false negatives.

When dealing with problems from clinical contexts, it is especially important to avoid false negatives, and sensitivity tells us how well our model is detecting the presence of each disease. Additionally, we want it to be specific, i.e., the rate of true negatives concerning false positives to be high, thus aiming for the best detection of healthy patients. Therefore, we computed the geometric mean of sensitivity and specificity (G-Mean) to select the best ECG classification model. It is computed as:

$$\text{G-Mean} = \sqrt{\text{Sens} \times \text{Spec}}. \tag{4.4}$$

The G-Mean was calculated from the average sensitivity and specificity between all classes, to guarantee equal relevance for all diseases. As such, under-represented classes are not overshadowed by better-performing and highly-represented categories.

Furthermore, we also computed the accuracy (Eq. 4.5) and precision (Eq. 4.6), to compare the performance of the 1D, 2D and multimodal approaches with the existing literature:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.5}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.6}$$

The epoch that presented the minimum validation loss was selected for extracting the final results.

Decision thresholds applied to the sigmoid activation's output were optimized by individual diseases, chosen by those that provided the best performance (highest G-mean) on the validation set.

4.3.4 Results

4.3.4.1 1D Approach

For each of the six approaches described in Section 4.3.3.2, the architectures' hyperparameters were optimized based on the validation loss, and the model with the optimized

Table 4.2: Classification results obtained with RNN and 1DCNN+RNN models. # Param represents the number of parameters of each model. The best results are highlighted in **bold**. Spec, Sens, and G-Mean denote specificity, sensitivity, and the geometric mean of these two values, respectively.

1D Strategy	# Param	Metric	MI	STTC	CD	HYP	AD	Total	G-Mean (%)
GRU	249 732	Sens (%)	84.81	83.37	78.71	72.24	77.11	**79.67**	**80.35**
		Spec (%)	78.82	83.96	86.31	71.74	87.20	81.04	
LSTM	794 628	Sens (%)	73.60	82.79	82.12	76.81	68.24	75.42	77.72
		Spec (%)	85.90	84.39	78.68	65.63	90.80	80.09	
BiGRU	399 620	Sens (%)	79.20	82.41	80.12	76.43	76.89	78.95	80.00
		Spec (%)	82.24	83.78	87.93	67.05	88.16	**81.07**	
BiLSTM	532 484	Sens (%)	78.84	84.89	78.71	71.10	75.68	78.18	79.51
		Spec (%)	82.17	80.79	87.33	70.05	87.12	80.87	
1DCNN+GRU	219 536	Sens (%)	72.88	84.13	67.47	66.54	69.55	72.33	75.28
		Spec (%)	78.88	75.30	83.78	70.63	86.16	78.35	
1DCNN+LSTM	81 296	Sens (%)	71.61	88.15	68.67	63.50	73.38	74.04	75.53
		Spec (%)	77.70	71.95	84.98	68.84	84.88	77.06	
GRUAtt	249 861	Sens (%)	83.72	80.88	76.71	78.33	69.11	76.55	78.33
		Spec (%)	80.50	86.16	85.89	63.53	89.44	80.15	

configuration was tested on the test set. The results obtained on the test set are presented in Table 4.2. The optimized architectures had the following configurations: (a) GRU and GRUAtt: 3 hidden layers, each with 128 hidden units and a dropout rate of 0, (b) LSTM: 2 hidden layers, a hidden size of 256 and a dropout rate of 0, (c) BiGRU: 2 hidden layers, a hidden size of 128 and a dropout rate of 0.5, (d) BiLSTM: 2 hidden layers, a hidden size of 128 and a dropout rate of 0.5, (e) 1DCNN+GRU: a hidden size of 256 and a dropout rate of 0.3, (f) 1DCNN+LSTM: a hidden size of 128 and a dropout rate of 0.3.

Overall, the best sensitivity value was achieved with the GRU model (79.67%), and the BiGRU model generated the highest specificity (81.07%). Considering the geometric mean of these two metrics, the GRU network outperforms all the remaining models (G-Mean of 80.35%).

In general terms, the use of 1D convolutional layers prior to the recurrent layers did not strengthen the network's performance. Also, considering the G-mean, the GRU-based approaches outperformed the LSTM-based ones, while simultaneously having a lower number of parameters. Comparatively, the performance of the GRU network did not improve when adding bidirectionality, while for the LSTM, the bidirectional version achieved slightly better results. The use of a self-attention mechanism did not enhance the GRU model's effectiveness. While there isn't, as expected, a direct relationship between the number of parameters and performance, it is clear that adding complexity to each different architecture does not necessarily improve results, particularly observing that neither BiGRU nor GRUAtt outperformed GRU.

4.3.4.2 2D Approach

Table 4.3 shows the classification results, for the test set, obtained for each CNN-based approach described in Section 4.3.3.3. The optimized architectures had the following configurations: (a) AlexNet: #filters=16, a batch size of 256, a learning rate of 0.01 and a dropout rate of 0; (b) ResNet: #filters=16, batch size of 128, a learning rate of 0.01; (c)

Table 4.3: Mean classification results obtained with CNNs (AlexNet, ResNet50, VGG16, MobileNetV2 and AlexNetAtt). # Param represents the number of parameters of each model. The best results are highlighted in **bold**. Spec, Sens, and G-Mean denote specificity, sensitivity, and the geometric mean of these two values, respectively.

2D Strategy	# Param	Metric	MI	STTC	CD	HYP	AD	Total	G-Mean (%)
AlexNet	46 334 468	Sens (%)	64.56	82.98	73.09	68.06	76.34	73.85	75.11
		Spec (%)	74.22	76.95	77.96	73.63	80.48	76.38	
ResNet50	1 494 356	Sens (%)	68.54	78.20	68.67	65.40	60.13	67.31	71.75
		Spec (%)	73.73	79.39	77.36	69.37	85.84	76.48	
VGG16	285 347 124	Sens (%)	67.81	84.51	67.07	66.16	79.63	**74.62**	75.19
		Spec (%)	72.17	75.49	80.00	73.21	79.04	75.77	
MobileNetV2	2 230 724	Sens (%)	57.68	63.86	57.83	32.32	83.57	65.05	73.03
		Spec (%)	75.40	82.50	85.71	90.95	71.20	**81.98**	
AlexNetAtt	46 335 880	Sens (%)	60.94	74.76	59.84	58.56	82.69	70.36	**75.95**
		Spec (%)	77.45	83.66	88.11	82.68	76.40	**81.98**	

VGGNet: #filters of 16, a batch size of 128, a learning rate of 0.1 and a dropout rate of 0.3; (d) MobileNetV2: #filters of 32, a batch size of 16, a learning rate of 0.1 and a dropout rate of 0; and, (e) AlexNetAtt: #filters of 8, a batch size of 16, a learning rate of 0.01 and a dropout rate of 0.

According to the results, the best G-mean (75.95%) was obtained with the AlexNetAtt model (Table 4.3). Additionally, the highest sensitivity (74.62%) and specificity (81.98%) were obtained with the VGG16 and MobileNetV2/AlexNetAtt models, respectively. Since we are interested in obtaining a model with both high sensitivity and specificity, the AlexNetAtt is the best model to perform ECG classification using these images. Moreover, the AlexNetAtt model has fewer parameters compared to the VGG16 model.

Overall, these results showed that the ECG to image transformations proposed in [7] which were applied to ECG signals corresponding to one heartbeat can also be applied to ECG signals of several heartbeats. The images contain not only the temporal details but also information regarding the relative amplitude of the different waves present in the ECG signal. Thus, CNNs can be used to extract features from these images to perform ECG classification.

4.3.4.3 Multimodal Fusion

Table 4.4 presents the test set results for each fusion approach, both with and without the additional attention layer. Among the tested hyperparameters in each strategy, the final model was selected from the lowest validation loss.

The optimized fusion architectures include the following configurations: (a) Late Fusion: final hidden layer size of 128 and a learning rate of 0.001; (b) Late Fusion with attention: final hidden layer size of 128 and a learning rate of 0.01; (c) Early Fusion: activations from the fifth convolution block, with a final hidden layer size of 128 and a learning rate of 0.1; (d) Early Fusion with attention: activations from the fourth convolution block, with a final hidden layer size of 256 and a learning rate of 0.01; (e) Joint Fusion: activations directly after the last convolution block, with a final hidden layer size of 256 and a learning rate of 0.01; and (f) Joint Fusion with attention: activations after the first

Table 4.4: Classification results obtained with the late, early and joint fusion approaches, with and without an additional attention layer following unimodal features concatenation. # Param represents the number of parameters of each model. In late and early fusion approaches, the count of trainable parameters is shown, with the sum of multimodal fusion weights and those from unimodal GRU and AlexNetAtt networks presented in parentheses, depending on the layer from which the activations were retrieved (early fusion). In the joint fusion approach, all unimodal and multimodal weights are adjusted during training. The best results are highlighted in **bold**. Spec, Sens, and G-Mean denote specificity, sensitivity, and the geometric mean of these two values, respectively.

Multimodal	# Param	Metric	MI	STTC	CD	HYP	AD	Total	G-Mean (%)
Late Fusion	1 668	Sens (%)	75.59	88.91	82.73	77.19	76.34	79.82	**79.42**
	(46 587 278)	Spec (%)	86.27	77.56	83.18	64.11	88.72	79.02	
Late Fusion (Attention)	1 884	Sens (%)	79.75	84.13	76.31	72.24	86.64	**81.53**	74.35
	(46 587 494)	Spec (%)	67.83	75.30	55.56	64.11	79.84	67.80	
Early Fusion	623 620	Sens (%)	80.47	85.47	79.12	68.44	74.15	77.93	78.80
	(1 057 032)	Spec (%)	76.40	79.15	84.32	73.11	88.40	79.68	
Early Fusion (Attention)	592 516	Sens (%)	73.06	77.82	78.31	71.48	79.41	76.87	76.20
	(878 216)	Spec (%)	68.39	80.12	82.70	66.11	83.52	75.54	
Joint Fusion	4 137 480	Sens (%)	70.71	83.37	71.29	72.62	71.52	73.67	76.56
		Spec (%)	81.18	78.11	87.03	67.21	88.16	79.55	
Joint Fusion (Attention)	40 712 200	Sens (%)	77.94	88.15	77.71	66.16	65.61	74.62	77.16
		Spec (%)	81.06	78.41	84.74	69.11	89.60	**79.79**	

fully connected layer, with a final hidden layer size of 128 and a learning rate of 0.001. Again, dropout and L2 regularization were both set to 0, and the *Adam* optimizer was employed.

The results of Table 4.4 show that the late fusion approach without a self-attention layer attained the highest performance on the G-Mean between the sensitivity and specificity of the test set, with 79.42%. However, this model did not individually achieve the highest sensitivity and specificity, which were instead attained by the late fusion model with attention (sensitivity of 81.53%) and the joint fusion model with attention (specificity of 79.79%).

This outcome is not intuitive, considering the differences in complexity. Approaches with not only more parameters but also access to latent space information from earlier activations did not lead to better performance. While the early and joint fusion approaches could have learned useful representations in the last common multimodal layers, the fusion of output probabilities led to the best performance.

Another key observation was the limited impact of the self-attention layer, which only improved results in the joint fusion model. However, this strategy, which taps into activations from a later layer, resulted in a tenfold increase in the number of parameters compared to other models. In all other strategies, incorporating self-attention led to a decrease in performance.

Furthermore, given that each network leverages the results from individual modality training, one might expect these to set the baseline for performance evaluation. Yet, this hypothesis does not hold up when comparing the results in Table 4.2. The fusion models neither significantly enhanced performance nor outperformed the most effective strategy, the GRU model. Therefore, our findings suggest that multimodal DL may not provide substantial benefits for ECG classification tasks, contrary to what might be expected.

Table 4.5: Final classification results for the best performing models from the 1D (GRU), 2D (AlexNetAtt) and Multimodal (Late Fusion) strategies. The best results are highlighted in **bold**. MM denotes the late fusion multimodal approach.

Approach	Metric	MI	STTC	CD	HYP	AD	Total
1D	Sensitivity (%)	**84.81**	83.37	78.71	72.24	77.11	79.67
	Specificity (%)	78.82	**83.96**	86.31	71.74	87.20	81.04
	G-Mean (%)	**81.76**	**83.66**	82.42	**71.99**	82.00	**80.35**
	Precision (%)	57.90	**62.37**	**63.23**	26.13	81.48	**58.90**
	Accuracy (%)	80.35	**83.82**	**84.56**	71.80	82.94	**80.69**
2D	Sensitivity (%)	60.94	74.76	59.84	58.56	**82.69**	70.36
	Specificity (%)	77.45	83.66	**88.11**	**82.68**	76.40	**81.98**
	G-Mean (%)	68.70	79.08	72.61	69.58	79.48	75.95
	Precision (%)	48.14	59.33	60.10	**31.88**	71.90	57.11
	Accuracy (%)	73.13	81.51	81.60	**79.75**	79.06	79.03
MM	Sensitivity (%)	75.59	**88.91**	**83.73**	**77.19**	76.34	**79.82**
	Specificity (%)	**86.27**	77.56	83.18	64.11	**88.72**	79.02
	G-Mean (%)	80.75	83.04	**82.96**	70.34	**82.30**	79.42
	Precision (%)	**65.41**	55.82	59.54	22.94	**83.17**	56.47
	Accuracy (%)	**83.54**	80.31	83.08	65.70	**73.50**	79.22

4.3.4.4 Comparison

The multi-label classification of CVD was approached in three ways: using the original ECG signal; leveraging RP, MTF, and GAF transformations; and considering both in multimodal fusion strategies. These data schemes were considered to evaluate the relevance of image transformations and multimodal fusion in CVD prediction. Furthermore, we evaluated different DL techniques and included attention-based strategies, to verify which attained the best results. Table 4.5 presents additional performance metrics for the best network of each data approach, i.e., the GRU in the time series sub-net, the AlexNetAtt in the image sub-net, and the late fusion strategies.

4.3.5 Discussion

The present work aimed to compare multiple DL approaches for CVD classification from ECG signals. Both 1D and 2D representations of the ECG signals were used as input for various DL architectures, firstly in independent models and secondly combined in a multimodal fusion approach. One aim was to understand if merging these two data formats of the same signal could enhance the classifier's performance since previous research has suggested that different signal representations might contain and highlight different information.

From the results section, the first thing that stands out when looking at the general results from Table 4.5 is that the 1D approach outperforms both the 2D and the multimodal approaches, including the G-Mean between sensitivity and specificity. One plausible reason may be related to their capacity to directly process the raw ECG signals, capturing intricate temporal patterns more effectively than image-based or combined data representations. Moreover, the inherent complexity in effectively integrating the modalities in multimodal approaches might have contributed to their relatively lower performance.

This complexity can arise from challenges such as the dominance of one modality, overfitting due to increased model complexity, or inefficiencies in how different modalities complement each other.

Additionally, 1D models also have the advantage of simplicity: they do not require the additional step of ECG-to-image transformation and have fewer parameters (Table 4.2) compared to the 2D models (Table 4.3). At first glance, these results might question the utility of Multimodal DL in ECG classification, particularly given the increased complexity without a corresponding enhancement in performance. Nonetheless, it is important to emphasize that sensitivity is of extreme importance in our context, as it is especially important to avoid false negatives when dealing with the detection of diseases. In that sense, the multimodal approach was the best one. The precision values were low for all models, indicating that from the total positive cases detected, only around 59% had the disease. Again, in a clinical context, a false positive is less problematic than a false negative. Nevertheless, further improvements on this matter would be beneficial to reduce the cases of false positive diagnoses in a real-world scenario.

As described in section 4.3.2.2, several studies have shown enhanced CVD classification accuracy when using CNN approaches after transforming 1D ECG signals into 2D images. The most commonly applied transformation is the time-frequency spectrogram using STFT, but a recent study concluded that for the detection of MI and arrhythmia cases, performing the RP, MTF, and GAF conversions was more advantageous, leading to better performances [7]. In this study, however, the 2D approach did not outperform the use of the raw ECG signal. Some factors that could explain these results are worth discussing. On the one hand, CNNs and RNNs have distinct strengths and weaknesses: while CNNs are more powerful in finding local spatial features, RNNs (namely those with gated structures like GRU and LSTM) are designed to handle long-term dependencies from sequential data. This characteristic might have been an advantage in the present context since we are dealing with multiple heartbeat signals and not dividing them into single beats, where the search for local patterns is probably of higher importance. Another aspect to consider relates to the applied transforms: these highlight the temporal dynamics within the ECG signal, such as its periodicity while losing some features more related to the signal's waveforms. Therefore, the conversion of the signal to an image using these transforms may be more beneficial if the disease is characterised by anomalies that are preserved and the opposite happens if the detection of the CVD relies on deviations in specific waveforms. At last, computational resources limited the direct usage of applied transformations, requiring an additional resizing from 1 000 values to 256. This process may have prevented some key patterns from being detected, although one should consider that using the original size could demand a very significant increase in model complexity.

Analyzing the results for each disease individually reveals distinct patterns in model performance. The multimodal approach generally achieves the highest sensitivity, particularly excelling in the STTC category (88.91%), demonstrating its efficacy in detecting

true positives in this condition. The 1D models also perform strongly across all categories, particularly in MI (84.81%), whereas the 2D strategy tends to fail in meeting high sensitivity standards. In terms of specificity, no single strategy consistently outperforms others, as this varies significantly among different diseases. However, when considering the geometric mean of sensitivity and specificity, both the 1D and multimodal approaches either lead or closely approach the best performance. Such conditions may be related to the fact that the late fusion benefits from previous learning of the 1D network, while the 2D strategy struggles to match these results. Regarding precision, no strategy can attain particularly high results, indicating the difficulty of the models in maintaining a low false positive rate, except for the unbalanced non-disease AD class (Table 4.1). Especially in the HYP class, which contains fewer examples, the maximum precision is only 31.88%, obtained with the 2D model. Accuracy levels are fairly similar across different strategies. Although it is widely used across literature and we have used it for comparison purposes, accuracy does not effectively reflect predictive capacity in imbalanced datasets, a common issue in diagnostic contexts.

The 1D model consistently demonstrates high performance across various metrics and conditions, highlighting its effectiveness and reliability in ECG classification. While the 2D model demonstrates some strengths, especially in specificity for HYP, it generally underperforms compared to the 1D and multimodal approaches. These findings suggest that multimodal methods may be particularly advantageous in certain conditions, while also emphasizing the robustness and versatility of 1D models in ECG classification. For practical applications, the choice of model (1D, 2D, or multimodal) for ECG classification should be informed by the specific cardiac condition being targeted, as each model exhibits unique strengths in different scenarios.

Previous studies using the PTB-XL dataset achieved an average accuracy and sensitivity of 76.5% and 66.2% [182]; an average accuracy of 76.3% [181]; a mean sensitivity, specificity and accuracy of 86.8%, 91.0%, and 89.0%, respectively [204]; and an overall AUC and sensitivity of 93.3% and 81.3%, respectively [214]. The results obtained in this work with both the 1D and multimodal approaches outperform those achieved in [182] and [181]. An average accuracy/sensitivity of 80.69%/79.67% and 79.22%/79.82% was achieved in the unimodal and multimodal strategies, respectively. Additionally, our models are simpler for practical use since they do not require manual feature extraction as done in [182] and [181]. The results obtained in [204] are comparatively superior regarding sensitivity and specificity. Nonetheless, it is important to note that these results were severely enhanced after the implementation of a class imbalance handling approach, where synthetic signals were sampled. Before applying this technique they had achieved a mean specificity and sensitivity of 58.8% and 92.8%, which indicates that our models exceeded their classification power when entirely based on the provided samples. Additionally, no discussion is provided in this work regarding train-test splits and testing using solely non-synthetic samples. Nevertheless, while we applied class weights in the loss computation to counteract the imbalance problem, using a synthetic data generation process before the

training could lead to improvements. The approach developed in [214] achieved a slightly higher sensitivity than that achieved in the present paper. However, it is important to analyze the computational cost that is associated with an improvement of $\sim 1\%$ in the overall sensitivity. In this work, the authors firstly propose a multi-view multi-scale network for extracting information from multiple ECG leads using a multi-scale one-dimensional CNN structure to extract features at different scales of the ECG signal. As the computational complexity of this first approach was very high, the authors proposed a smaller single-view network. Nevertheless, when comparing to our GRU network, the number of parameters of the network is still higher, since ours has $\sim 2.5 \times 10^4$ parameters and the multi-scale network has $\sim 3.9 \times 10^4$ parameters. Furthermore, the authors did not provide the specificity value, which may be much lower than the sensitivity value. For instance, if we did not take the G-mean into account but only sensitivity, our best-performing model would be the multimodal late fusion (with attention) model, which only achieved a specificity of 67.80% while achieving a sensitivity of 81.53% (Table 4.4). This emphasizes how crucial it is to take into consideration the G-mean of sensitivity and specificity in disease classification tasks.

4.4 Final Remarks

A key takeaway from this study is that higher complexity or providing more detailed inputs does not necessarily lead to better performance. In fact, the approach that ultimately achieved the best performance was the least complex one, highlighting the importance of balancing complexity and effectiveness. One plausible explanation for this outcome is that 1D models are well-suited to directly process raw ECG signals, effectively capturing intricate temporal patterns without the need for additional transformations. On the other hand, image-based and multimodal representations introduce additional layers of complexity that may not always contribute to improved classification performance.

The challenges associated with multimodal approaches, such as effectively integrating different data representations, might have contributed to their relatively lower performance. Issues such as modality dominance, where one representation overshadows the contribution of the other, and the increased risk of overfitting due to greater model complexity, may have hindered their effectiveness. Additionally, inefficiencies in how the two modalities complement each other could also explain the performance gap. These findings reinforce the notion that simplicity in model design can often yield robust and reliable results, aligning with one of the main goals of this thesis - to explore the trade-off between model complexity and generalization, while advocating for lower complexity as a goal in itself due to the environmental and computational costs associated with large DL models.

However, interestingly, the multimodal approach achieved the best sensitivity among the evaluated methods, which is a crucial factor in scenarios where identifying positive cases is of utmost importance. It is also important to consider that the reported results

represent an average across different cardiovascular conditions. A closer analysis reveals variations in performance across specific conditions, suggesting that different diseases may benefit from different processing strategies. For example, while the 1D model consistently demonstrated high performance across various metrics and conditions, the 2D model exhibited notable strengths in terms of specificity for HYP.These findings suggest that multimodal methods could hold advantages in particular conditions, yet the robustness and versatility of 1D models was again evidenced for ECG classification. Future improvements could potentially be achieved by processing each image representation independently in separate models and subsequently merging the outputs, rather than combining them at the input level. Such an approach might allow each representation to contribute its strengths without introducing unnecessary complexity at earlier stages of processing.

Overall, this study highlights that increasing model complexity and incorporating more detailed data does not always lead to improved outcomes. In biosignal processing, while certain scenarios may gain from added dimensionality or the use of multimodal approaches, their practical application must be carefully considered in light of the challenges they pose, especially regarding generalization and computational efficiency.

5

Deep Learning applied to ECG Noise Detection and Removal

This chapter explores the application of Deep Learning (DL) methods for noise handling in ECG records, with a primary focus on noise removal. Various approaches were tested to assess their effectiveness, and a noise detection and classification method is also proposed. The models were trained using data from the PTB-XL dataset, previously described in Section 4.2, now combined with noise samples from the MIT-BIH Noise Stress Test database. A key objective in this chapter is to analyze the generalization power of models trained on data acquired in controlled clinical settings to more complex, real-world occupational environments, specifically within a factory assembly line.

5.1 Introduction

Wearable technologies have extended biosignal acquisition beyond clinical walls, embedding sensors into the daily life of a wide part of the population to track heart rate, sleep patterns, and physical activity in real-time. Wearables also enable clinicians and researchers to take ECG monitoring out of the clinical and laboratory environment to assess cardiovascular health at virtually any environment and moment in time. As a result, biosignals are now being used in a wide range of applications in addition to clinical applications. Monitoring occupational health through ECG can yield positive outcomes at both individual and organizational levels, as work-related cardiovascular disease (CVD) have direct implications on productivity, contributing to higher absenteeism and reduced efficiency [4, 184].

However, while acquiring ECG signals with wearables is relatively straightforward, ensuring data quality and readability remains a critical challenge. In non-clinical, uncontrolled environments, freely moving subjects and various noise sources can significantly distort the acquired signals, making accurate analysis and interpretation difficult. The most common sources of noise in ECG signals originate from Muscle Activation (MA) near the electrode placement, Electrode Motion (EM), and Baseline Wander (BW) [86],

which are the primary focus of the present work. Artifacts that overlap with the frequency spectrum of the ECG signal (~[1 Hz,45 Hz]) are particularly challenging to remove, as these cannot be removed with linear filters (low-, band-, nor high-pass filters) [104].

The impact of noise is particularly pronounced in occupational environments where movements are more frequent and intense. Work settings like factories, where work is often repetitive and performed in awkward postures, lifting of potentially heavy components, and under high levels of noise and stress, exemplify environments where physiological monitoring should be a standard practice, as argued in section 2.3. However, these environments are also highly susceptible to noise interference, since workers engage in rapid, repetitive, and abrupt movements. In such scenarios, factors like muscle contractions around the electrodes and minor electrode shifts are more likely compared to clinical settings, where subjects are typically in a resting state during examinations. Regardless of the context - clinical or occupational - ensuring data quality is crucial for deriving reliable information. Effective noise removal and quality assessment techniques are essential to guarantee accurate data interpretation and decision-making.

5.2 The MIT-BIT Noise Stress Test Dataset

MIT-BIH Noise Stress Test Database [136] is a publicly available dataset from PhysioBank [53] which contains records of EM, BW and MA noise. The noise dataset used contains three half-hour noise recordings that are typically found in ambulatory ECG recordings. These recordings were obtained using physically active volunteers, with standard ECG recorders, leads, and electrodes that were placed to ensure that the recording predominantly captured noise rather than ECG signals. Two channels of noise recordings were collected at 250 Hz.

BW is a low-frequency noise usually caused by respiration or body movements that lead to slow changes in electrode-skin impedance consequently causing the ECG baseline to drift [94]. EM artifacts occur due faster changes in the electrode-skin impedance, usually caused by abrupt physical movement of the electrodes relative to the skin. These artifacts typically have a frequency range of 0.5 to 10 Hz and can manifest as large-amplitude waveforms in the ECG signal, potentially leading to misinterpretation [86]. MA noise originates from the electrical activity of surrounding skeletal muscles which cause high-frequency fluctuations (EMG) to be superimposed on the ECG signal. Its frequency range usually overlaps with that of the QRS, ranging from 5 to 50 Hz, making it more difficult to isolate from the signal [104].

5.3 Automatic Noise Removal from ECG signals[1]

5.3.1 Objectives

The primary objective of this work is to develop and evaluate a DL model capable of effectively removing the three most common types of noise - MA, EM, and BW - from ECG signals. While the model was trained using the PTB-XL database and noise samples from the MIT-BIH Noise Stress Test database, a key goal of this study is to assess the model's generalization beyond controlled clinical environments. To this end, data was collected in a real-world occupational setting under the scope of the *OPERATOR* - MIT Portugal Project. The data was acquired with a single lead wearable device, in an automotive assembly line, where subjects performed routine tasks involving a wide range of motor strategies, resulting in complex noise interference. Evaluating the model's performance in such an environment provides valuable insights into its potential for practical deployment in workplace health monitoring applications.

The specific objectives of this study can be summarized as follows: (i) developing a lightweight GRU-based DL model that balances performance and computational efficiency, (ii) utilizing a diverse and heterogeneous dataset to ensure the model's ability to handle ECG signals from a wide range of patients, including those with different cardiovascular conditions, (iii) testing the model in a challenging real-world industrial setting, offering an evaluation of its generalization to uncontrolled environments.

5.3.2 Related Work

Regarding state-of-the-art approaches in ECG signal denoising, we can distinguish traditional approaches from DL-based ones. Traditional noise removal techniques encompass empirical mode decomposition, wavelets, sparsity-based models, bayesian filters, non local means denoisers, and hybrid models [31, 197]. In comparison to these, deep neural networks have shown enhanced performance in ECG denoising, having the advantage of being adaptable to different noise amounts and sources, if trained with sufficient and relevant data. Most ECG denoising approaches that employ DL focus on denoising autoencoders (DAE), for which several model architectures have been proposed, including Feed Forward Neural Network (FFNN), Convolutional Neural Network (CNN), and Long Short Term Memory (LSTM) cells. Rodrigues and Couto [157] presented a feed forward NN with three hidden layers for the reduction of electrode motion artifacts. With their model, they showed that a variety of QRS detectors perform better after using it to denoise ECG signals. In [121], two fully-connected DAEs, one with a single hidden layer (SHL) and one with multiple hidden layers (MHL) were trained on ECG signals to which Gaussian white noise was added.

[1]This section is based on the publication "Cleaning ECG with Deep Learning: a denoiser tested in industrial settings" by **Mariana Dias**, Phillip Probst, Luís Silva, and Hugo Gamboa, published in the journal SN Computer Science (2024).

Arsene, Hankins, and Yin [12] proposed two models for ECG denoising. The first was a CNN model consisting of six blocks that contain a convolutional layer followed by batch normalization layer, a rectified linear unit layer, and average pooling layer. The second was an LSTM model composed of two blocks that include an LSTM layer with 140 neurons followed by fully-connected rectified linear unit layer, also with 140 neurons. Both models use an input layer of 30000 neurons which is equal to the number of samples in the ECG input signal and a fully-connected regression output layer that reconstructs the original signal. To train and test their models they used two synthetic datasets and a real ECG dataset. In the synthetic datasets they added random and drift noise, while for the real ECG dataset a focus was put on motion artifact noise. The two models were compared with a traditional wavelet approach, showing that the CNN model outperformed the LSTM and the wavelet approach, when comparing the root mean square error between the reconstructed and original signal.

Antczack [9] developed an architecture consisting of an LSTM layer with 140 neurons followed by two ReLU layers of 64 neurons each that was trained on ECG sequences with a length of 600 samples. Their network was pre-trained with synthetic data to find an optimized architecture and subsequently fine-tuned with real data. Both datasets were corrupted by white noise. They compared their network to a wavelet approach and a bandpass filter and their model outperformed these approaches. Pre-training with synthetic data helped the model to converge faster when being fine-tuned.

A DAE that combines CNN and LSTM was proposed by Dasan and Panneerselvam [39]. The encoder of their model consists of eight convolutional layers of with max-pooling layers in between. The final component of the encoder consists of a LSTM cell followed by a fully-connected layer. The decoder mirrors the convolutional layers of the encoder with the difference that no batch normalization is applied and max-pooling is replaced by 1D up-sampling. They evaluated their model by comparing it to other convolutional as well stacked DAEs when applying white gaussian noise to ECG signals, showing that their approach outperformed the comparative networks.

An approach using a generative adversarial network (GAN) was proposed in [206]. The generator of the GAN receives signals corrupted with noise and is set to recreate the original noise-free signal. It consists of three fully-connected layers; the input and output layers are of size 310 which corresponds to the number of samples in the signals fed to the network. The generator of the network is trained with different loss functions that are composite combinations of the default generator loss, a distance function, a maximum local difference, and a mean square error. The discriminator receives either the output of the generator or a real noise-free ECG signal. It is set to distinguish if the provided input is either a real noise-free signal or a denoised signal.

While the presented approaches are all valid in their own right, they have certain limitations. With the exception of [121] and [9], all presented papers used either synthetic data and/or the MIT-BIH Arrhythmia database [135], which consists of data from only 47 subjects. There is the possibility that these networks are biased towards arrythmia

signals and their denoising results would not translate to other pathologies. Furthermore, in all works noise was applied to the entire sample, which is not necessarily representative of real-world scenarios (e.g., motion or muscle artifacts do generally appear for short amounts of time, unless the subject is moving constantly). Finally, while [121] and [9] used the PTB-XL database they only applied white noise to their ECG samples. This shows that there is a lack of models that are trained with (1) more diverse datasets (i.e., containing more pathologies and data from healthy subjects), (2) realistic noise characteristics and (3) that are tested and evaluated on data collected in out-of-laboratory environments.

The methodology of the present study is divided into two parts. The first part focuses on the development and training of a noise removal model using ECG data from the PTB-XL dataset combined with noise samples from the MIT-BIH Noise Stress Test database. The second part evaluates the model's performance in a real-world industrial setting, assessing its effectiveness in denoising ECG signals collected from workers in an automobile assembly line.

5.3.3 Methodology

5.3.3.1 Part 1: Model development

Data pre-processing For developing the present work and training our model, we used two databases: the PTB-XL database [205] and the MIT-BIH Noise Stress Test database [136], both publicly available in PhysioBank [53]. PTB-XL is described in the previous section (section 4.2) and the MIT-BIH Noise Stress Test database contains three half-hour recordings of BW, MA, and EM noise. Two channels of noise were collected at 250 Hz, by placing the electrodes in such a way that the ECG was not visible.

The following pre-processing steps were applied to the ECG signals: (i) the 500 Hz version of the PTB-XL database was down-sampled to 360 Hz (same as the MIT-BIH arrhythmia database); (ii) a second-order bandpass butterworth filter was applied to all ECG signals, with a high-pass cutoff frequency of 1 Hz and a low-pass cutoff frequency of 45Hz to eliminate high-frequency noise [104]; and (iii) a *Min-Max* normalization was performed (normalization based on the minimum and maximum values of each record). The noise data was also resampled to 360 Hz.

Training, Validation, and Test Sets The ECG data from PTB-XL was divided in training, validation, and test sets, with a proportion of 70%, 15%, and 15%, respectively. For each signal, the three leads with the lowest number of peaks were selected and, for the training set, the 3 leads were used as separate signals; for the validation and test sets, 1 of the 3 leads was randomly selected. The rationale of this procedure is: as all leads are records of the same heart activity, a higher number of peaks in one lead is likely due to the presence of noise. This way, we had 45777 (15259×3 leads) ECG samples for training, 3270 for validation, and 3270 for testing. Regarding the noise data, both provided channels were

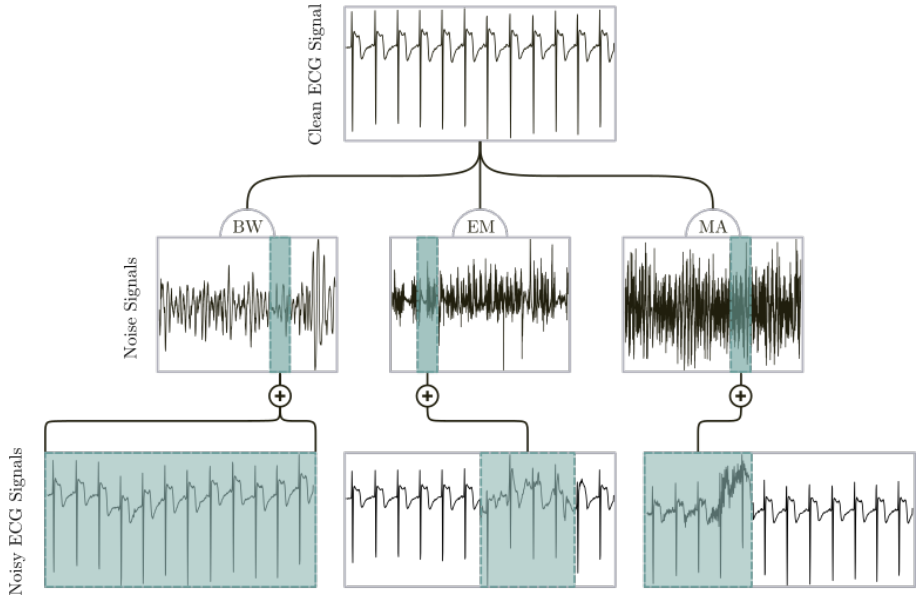


Figure 5.1: Schematic representation of the creation of noisy samples from adding MA, EM and BW noise to the clean ECG samples from PTB-XL.

used, totaling 60 minutes of BW, MA, and EM noise. 80% (48 minutes) was used for training, 10% (6 minutes) for validation, and the remaining 10% for testing.

The model's input: noisy samples For training our model, we added to each 10-second clean ECG signal a random portion of noise, with a length of 2 to 6 seconds. The EM and MA noises were added to a random part of the clean ECG signal while the BW noise was always added to the entire 10 seconds, as illustrated in Fig. 5.1. Additionally, to some records, mixed samples of different types of noise were added. The set of all possible combinations was: (MA, EM, BW, BW + MA, BW + EM, MA + EM, BW + MA + EM). There was a probability of 1/3 that the added noise was MA, of 1/3 that the added noise was EM and of 1/3 that the added noise was either BW or any combination. The reason for this is the fact that BW noise is the easiest to remove and having combinations of different noise simultaneously is less likely in the real world scenarios. In the training set, since we used 3 leads from the same signal separately, a different type of noise was added to each lead. In the training and validation sets, in order to include different proportions of noise, the noise was multiplied by a randomly chosen factor between [0.7, 1.5] before being added to the ECG signal. These boundaries were chosen to ensure that the model was exposed to neither too low nor unrealistically high amounts of noise. In the test set, the added noise had known SNR_{in} values of either 0, 5, 7, or 10 dB.

Architecture definition and training During the development of the GRU model, various architectures were tested in the validation set. The optimized hyperparameters are presented in Table 5.1. The model was fed with ECG signals following pre-processing and noise addition steps. Prior to inputting the data into the model, a Min-Max normalization

Table 5.1: Hyperparameters of the model and corresponding optimized values.

Hyperparameter	Tested values
Number of layers	{2,3}
Number of neurons	{32, 64, 128, 256}
Dropout rate	{0, 0.3}
Bidirectional	{True, False}

was applied, resulting in sequences of 3600 samples each (representing 10 seconds recorded at 360 Hz) with values ranging from 0 to 1. The desired output consisted of the clean signals after pre-processing. While the model's number of layers, neurons, and the dropout rate (applied to the output of GRU layers, except for the last one) were subject to optimization, certain parameters of the model were preestablished. In the bidirectional approach, only the first or first two layers (in the cases where there were three) were configured to be bidirectional. The loss function employed was the Root Mean Square Error, measuring the discrepancy between the predicted and desired outputs. All models were trained using the $Adam$ optimizer, for 130 epochs with a batch size of 256 and an initial learning rate of 0.005. For each approach, the model parameters from the epoch with the lowest validation loss during training were saved. Finally, validation losses were compared and the model achieving the lowest value was tested on the test set.

We conducted model training using a Nvidia GeForce GTX 1080 Ti GPU and the code was implemented in the DL framework Pytorch [146].

Evaluation metrics To evaluate model performance on the PTB-XL database, the main metrics found in the literature were used: the Root-Mean-Square Error (RMSE), the improvement in the Signal-to-Noise Ratio (SNR_{imp}), and the Percentage-Root-Mean-Square Difference (PRD) [31]. Equations 5.1 to 5.5 describe these metrics, where y is the original clean ECG signal, $\tilde{y}$ is the noisy signal, and $\hat{y}$ is the denoised signal. It is important to note that all these metrics imply the existance of a clean ground truth signal to compare the denoised output with, which means that they are not useful for real use cases. The equations that define these metrics are the following:

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1}[y(n) - \hat{y}(n)]^2} \tag{5.1}$$

$$SNR_{imp} = SNR_{out} - SNR_{in} \tag{5.2}$$

$$SNR_{in} = 10 \times log_{10}\left(\frac{\sum_{n=0}^{N-1}[y(n)]^2}{\sum_{n=0}^{N-1}[\tilde{y}(n) - y(n)]^2}\right) \tag{5.3}$$

$$SNR_{out} = 10 \times log_{10}\left(\frac{\sum_{n=0}^{N-1}[y(n)]^2}{\sum_{n=0}^{N-1}[\hat{y}(n) - y(n)]^2}\right) \tag{5.4}$$

$$PRD = \sqrt{\frac{\sum_{n=0}^{N-1}[\hat{y}(n) - y(n)]^2}{\sum_{n=0}^{N-1}[y(n)]^2}} \times 100\% \tag{5.5}$$

5.3.3.2 Part 2: Testing the model on industrial setting data

The main goal of the present study is to build a model that is effective in the task of removing noise from ECG data acquired with wearable sensors in real-world scenarios, namely during work in a factory, where noise sources are various and extremely difficult to avoid. As such, After developing our model, we performed a test on data acquired in an industrial setting, with the aim of testing the robustness of the model. In section 5.3.3.2, the data acquisition protocol is described; in section 5.3.3.2 the applied method for testing the model with these data is defined; and in section 5.3.3.2 we justify the parameters used for evaluating its efficacy in the absence of clean ground truth data.

Data acquisition protocol A total of 46 subjects working in an automotive assembly line participated in the study. Data from 3 subjects were lost due to equipment problems during the acquisition, resulting in a final sample size of 43 participants (Table 2 contains the demographics information of the participants). Each session had a duration of 20 minutes to 1 hour where the workers performed their regular work tasks (it was not possible to standardize the time duration of the acquisitions because the regular assembly line functioning could not be compromised). The data collection was performed using the acquisition software *OpenSignals* (Plux, Lisbon, Portugal). A single-lead electrocardiography sensor with triode configuration (Plux, Lisbon, Portugal) was placed on the operator's chest, as represented in Fig. 5.2, and the sensor was acquiring with a sampling rate of 350 Hz. The electrode placement was chosen to minimize the muscle activation noise, once the workers are performing intense upper body work and there was the need of reducing the risk of obtaining irreversibly noisy data. In addition to the ECG, other sensors were used to acquire electromyography, respiration and movement data, nonetheless, this data was not used for the present study. The protocol was clearly explained to each participant, and all subjects gave their informed consent for inclusion before they participated in the study. The study was conducted in accordance with the Declaration of Helsinki, and the protocol was approved by the Ethics Committee of University of Porto.

The acquired raw data were stored and visually inspected. From the visual inspection, it was possible to see that, although noisy as predicted, the data was in general of satisfying quality considering the conditions under which the acquisitions were performed.

Pipeline for ECG denoising The main pipeline that was applied to the ECG data for testing our model can be described by the following sequence of steps: (i) randomly selecting 10-min from each subject's data, (ii) preparing the signal, which includes resampling to 360 Hz, and *MinMax* normalization; (iii) running the model giving the prepared signal as input; (iv) post-processing - the *post-alignment* step - which is the application of a 3rd

Table 5.2: Participants' demographics data.

Sex	4 F 39 M
Age	38.0 ± 7.7 yrs
Height	176.7 ± 6.9 cm
Mass	77.9 ± 11.1 kg

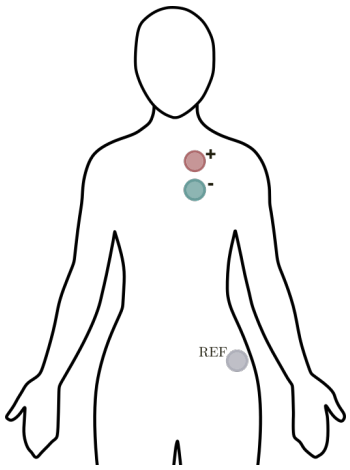


Figure 5.2: An illustrative representation of the ECG electrode placement.

order highpass butterworth filter with a cutoff frequency of 0.5 Hz. The last step was added to our pipeline due to a bias that the model was adding to the output when long signals were given as input, causing a misalignment.

For time optimization purposes, we evaluated the impact of segmenting the ECG signal prior to its use as input to the model. The segmentation process, depicted in Fig. 5.3, can be outlined as follows: (i) dividing the signal into 31-second segments with a 1-second overlap with the preceding one (excluding the initial 30-second segment), (ii) applying denoising to each segment using the BiGRU denoiser, and (iii) reassembling the segments. To reconstruct the signal, the initial 0.5 seconds of each output segment are cut, and the remaining overlapping 0.5 seconds undergo mean overlap computation, as indicated in green in Fig. 5.3.

Model evaluation on industrial setting data Testing on real-world data is a necessary step to understand if a model has the generalization power to properly execute the task it was designed for or if it only learned the features of the data it was trained with (*i.e.*, whether it is overfitted or has good generalization power). Nonetheless, this evaluation is not straightforward, since there is no ground truth to compare our results with. The literature on ECG quality assessment can be divided in 3 main approaches [87]: the first uses statistical metrics to assess if the signal is of good quality or not, based on the expected shape of the histogram of a clean ECG signal; the second is based on distance metrics from template signals; the third comprises machine learning models that perform classification distinguishing noisy from clean signals. The first two approaches fall short when we are

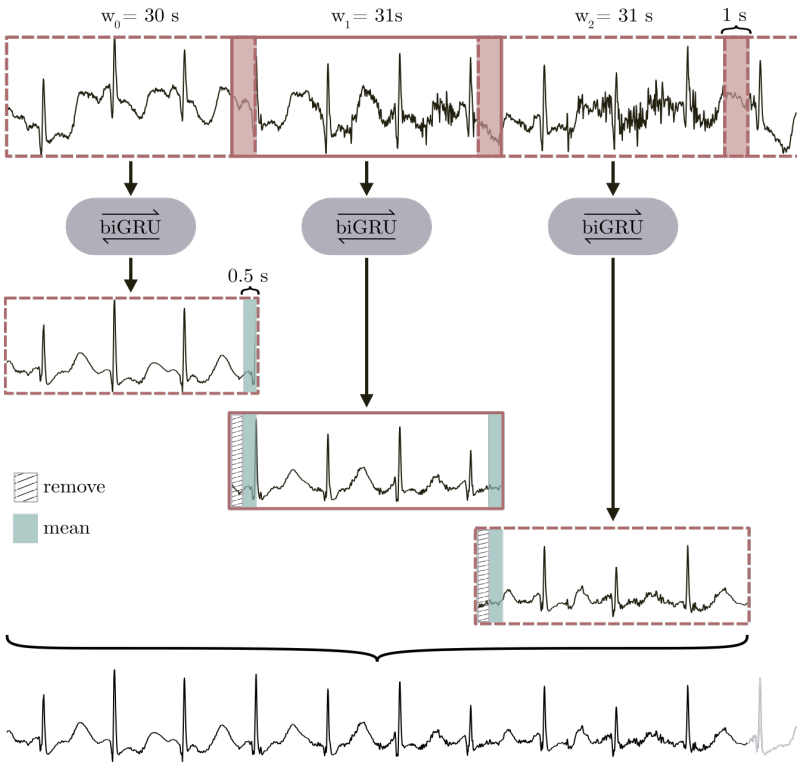


Figure 5.3: Illustration of the segmentation process applied to long ECG signals. Within the reconstruction phase of the process, "remove" and "mean" correspond to two key actions performed on output signal segments: "remove" indicates the portions of the signal that are cut, while "mean" designates the segments overlapped to compute the final reconstructed signal, depicted at the bottom of the figure.

dealing with data that are not acquired in a clinical setting, because the shape of the ECG signal and, consequently, of the histogram depend on the placement of the electrodes while acquiring the ECG. This way, clean ECG signals can have high distance values from ECG templates or have different histogram shapes not meaning that it is due to the presence of noise. The most recent studies on ECG quality assessment methods focus on distinguishing noisy from not noisy segments of signal and not on quality quantification [19] [216]. This can be useful for automatic deletion of noisy data (for example, for clinical purposes) or for selecting the parts of the data which should be denoised. For the purpose of this paper, nonetheless, it is not applicable as we are dealing with data that is almost continuously subject to noise and we are trying to improve its quality in order to allow the study of cardiovascular response in workers using wearables. Moreover, it's crucial for these algorithms to be made publicly available, enabling wider testing and application, and this is frequently not the scenario [19].

Previous results obtained within the scope of the OPERATOR project have shown a significant correlation between heart rate variability (HRV) and fatigue state of subjects performing repetitive work in a laboratory setting [30, 29]. These results suggest that from HRV we can obtain meaningful information from a subject's physiological state during work. Nonetheless, the various sources of noise that were present in an industrial

workstation contaminate the ECG data and, hence, lead to higher difficulties while performing R-peak detection for the computation of HRV metrics. As such, in order to evaluate the performance of the BiGRU model in this new dataset, the chosen approach was to evaluate the improvement in R-peak detection after cleaning the noise with our model, similarly to the strategy implemented by Rodrigues and Couto in [157].

In summary, for evaluating the model, we used a random 10-minute segment from the data of each participant and computed the number of missing peaks (MP) and wrongly detected peaks (WP) before and after performing the noise removal. To do that, we did the following steps: (i) R-peak detection, for which we employed a widely used Python algorithm from a well-know toolbox named *Neurokit* [118] (we used the default method, where the QRS complexes are detected based on the steepness of the absolute gradient of the ECG signal); (ii) compute the distance between each two consecutive peaks; (iii) compute the upper and lower limits for outlier detection, based on the 25% (Q1) and 75% (Q3) quartiles (and the Inter-Quartile Range (IQR)), as represented in equations 5.6 to 5.8; (iv) compute the derivative of the distances vector and check if it is higher than a minimum value of 50 (experimentally determined). The last criterion was added to make it robust to large (yet physiologically acceptable) heart rate fluctuations, avoiding erroneous outlier detections.

$$IQR = Q3 - Q1 \tag{5.6}$$

$$Upper\ limit = Q1 + 1.25 \times IQR \tag{5.7}$$

$$Lower\ limit = Q3 - 1.25 \times IQR \tag{5.8}$$

5.3.4 Results

5.3.4.1 Part 1: Model development

From the models described in section 5.3.3.1, the one that achieved the lowest RMSE on the validation set was the bidirectional GRU with 2 layers of 64 hidden units with 0 dropout rate. This model has 26121 parameters and achieved the best performance on the validation set after training for 58 epochs. Fig. 5.4 illustrates the evolution of the model performance throughout the training phase: after the first epoch, the model outputs a timeseries that is almost entirely the same as the input; after 10 epochs it gets closer to the clean signal and after training for 58 epochs the noise is essentially removed from the input signal. Fig. 5.5 shows some examples of the performance of the final model on different signals from the test set, corrupted with each type of noise and combination of different noises.

Table 5.3 demonstrates the mean results over all test signals according to the SNR_{in}. When higher amounts of noise are added to the input (i.e., lower SNR_{in}) the SNR_{imp} is higher. However, PRD and RMSE increase, indicating that the output has a higher denoising error.

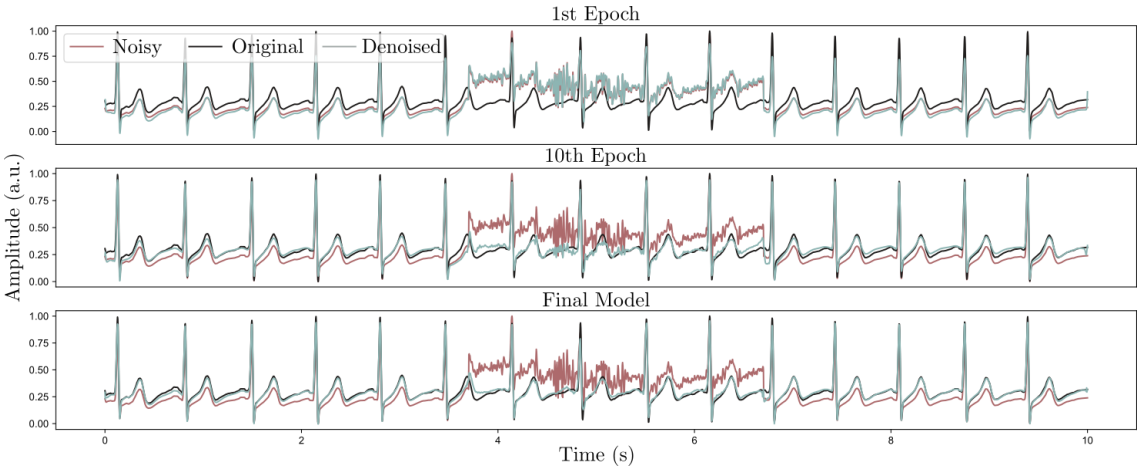


Figure 5.4: Denoiser perfomance after the first, the tenth and the 58th (best) epochs.

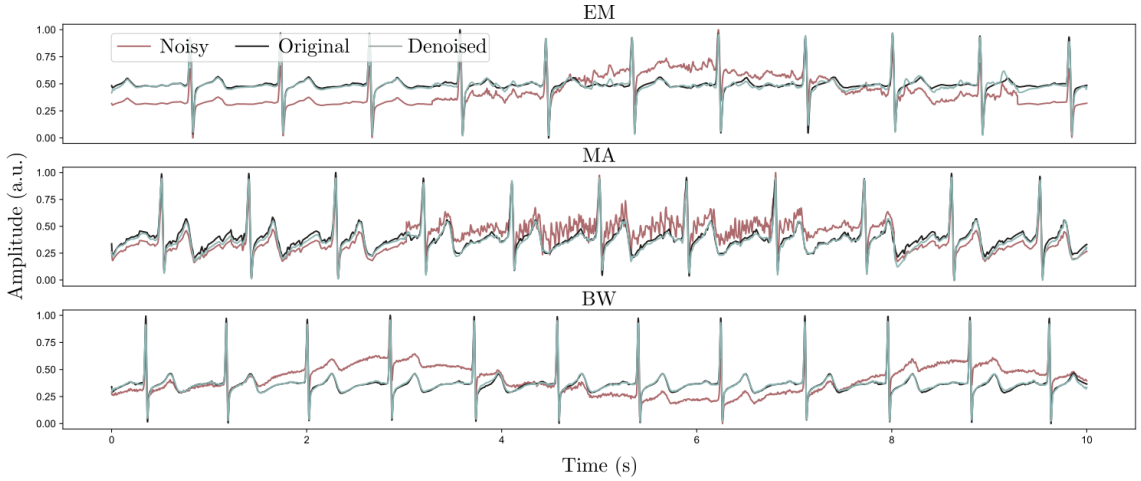


Figure 5.5: Denoiser perfomance on samples from the PTB-XL test set: EM, MA, and BW. Right: samples corrupted with different combinations of noise.

Regarding the model's performance on different types of noise, described in Table 5.4, the results show that the model is most and least effective in cleaning BW and EM, respectively.

In Fig. 5.7 we compare our model's performance with two traditional denoising methods: the *Neurokit* filter (a 0.5 Hz high-pass butterworth filter of 5th order, followed by powerline filtering) and the Pan-Tompkins algorithm. These denoisers are open source programs which are publicly available in the Neurokit Python Toolbox [118]. The RMSE

Table 5.3: Mean results obtained (SNR_{imp}, PRD, and RMSE) for each SNR_{in}.

SNR_{in} (dB)	SNR_{imp} (dB)	PRD (%)	RMSE
0	21.7	8.9	0.041
5	19.5	6.5	0.029
7	18.6	5.7	0.025
10	16.5	5.4	0.023

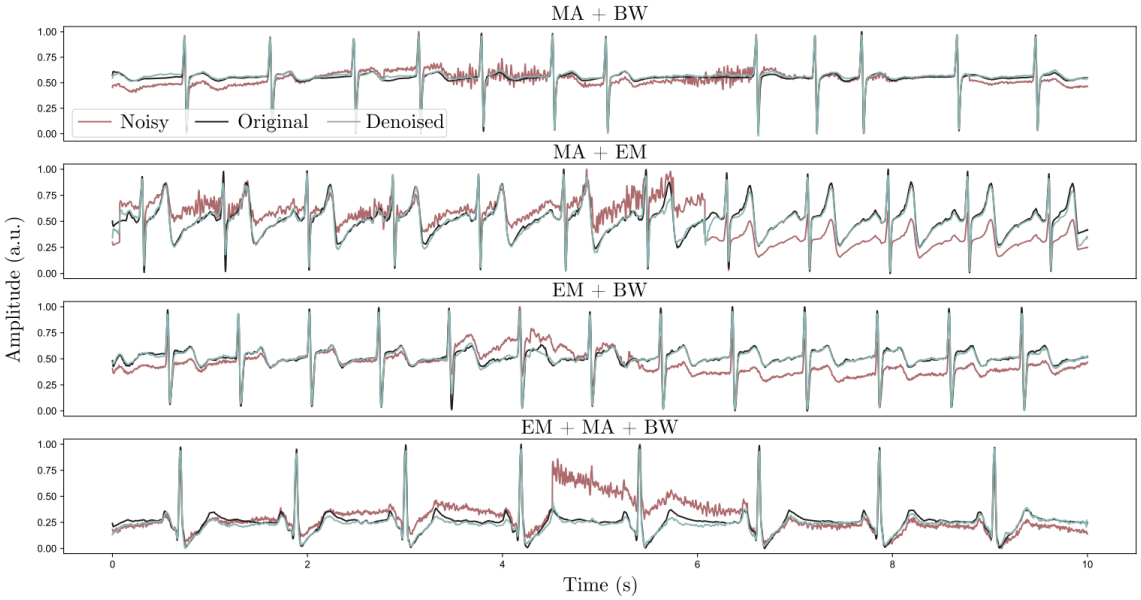


Figure 5.6: Denoiser perfomance on samples from the PTB-XL test set: samples corrupted with different combinations of noise.

Table 5.4: Results obtained (SNR_{imp}, PRD, and RMSE) per noise type.

Noise Type	SNR_{imp} (dB)	PRD (%)	RMSE
MA	19.2	6.4	0.029
EM	18.2	7.0	0.032
BW	20.6	5.5	0.024
MA + BW	19.4	6.5	0.027
EM + BW	19.5	6.1	0.027
MA + EM	19.2	6.4	0.029
MA + EM + BW	18.9	6.6	0.029

is 0.029, 0.393, and 0.405 when using the proposed GRU model, the Neurokit and the Pan-Tompkins denoisers, respectively, which shows that our model clearly outperforms these traditional denoisers.

5.3.4.2 Part 2: Testing the Model in Industrial Setting Data

As explained in Section 5.3.3.2, the results from the data acquired in the industrial work setting were obtained by segmenting the input signals. This was done to decrease the runtime of the model pipeline. Fig. 5.8 reveals the generated output when performing versus not performing segmentation around an example of a segmentation limit, showing that the results are similar, having little influence in the signal morphology around the segments limits. In the subsequent figures of the present section, all the denoised signals correspond to outputs obtained using the segmentation step.

The peak detection process, before and after denoising the signal, is represented in Fig. 5.9. This image illustrates the presence of outliers in the BPM signal, which was computed from the distance between each two consecutive peaks divided by the sampling frequency,

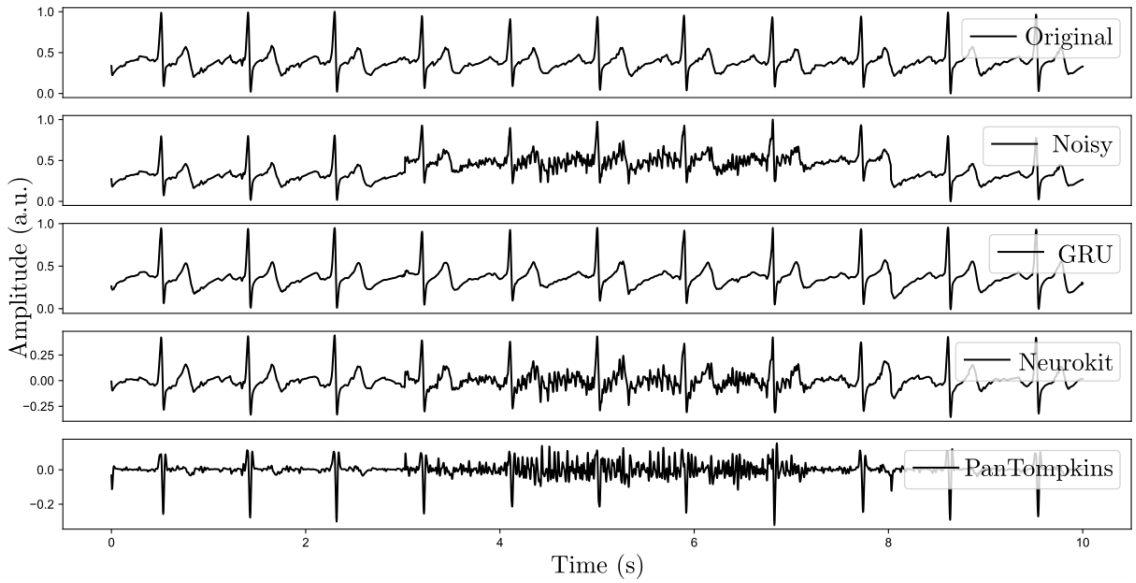


Figure 5.7: Comparative results of the proposed denoiser (GRU) and two denoisers publicly available through the *Neurokit* Python toolbox [118]: *Neurokit* and Pan-Tompkins.

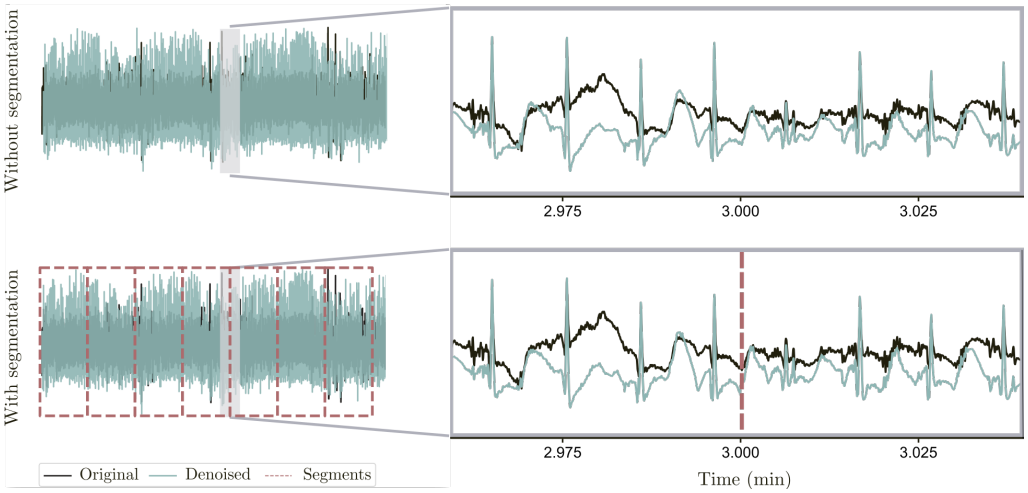


Figure 5.8: Comparison of the output signal around a segment limit when the segmentation of the signal was not performed (top) versus when it was (bottom).

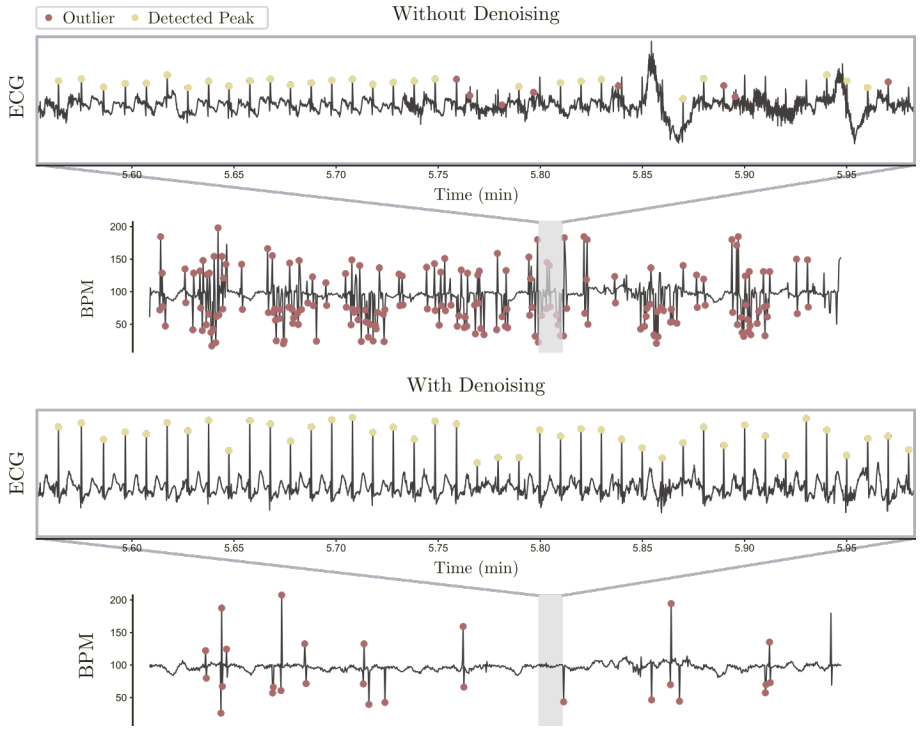


Figure 5.9: Presence of outliers in the beats per minute (BPM) plot, from the peak detection process in the ECG signal, before and after signal denoising.

360Hz, multiplied by 60s/min. The peak detection performance is extremely improved after applying the proposed algorithm.

Fig. 5.10 exhibits an example of the output of the model in comparison with the noisy input signal. In Table 5.5, we present the mean results across all 43 subjects, including the counts of missing and wrongly detected peaks (MP and WP, respectively) before and after denoising, along with the corresponding relative improvements (%Imp) achieved after signal cleaning using the proposed algorithm. These results distinguish between outcomes obtained before and after the implementation of the segmentation step. On average, more than 70% of the missing peaks are correctly detected after denoising the signal. Concerning WP, more than 50% of the peaks that were erroneously detected are corrected. These results demonstrate that the efficacy of the model does not change considerably due to the segmentation step. Nevertheless, performing signal segmentation significantly reduced processing time by about 50% for 10-minute sequences.

Table 5.5: Results obtained in R-peak detection before and after applying the proposed model. Two different approaches (with and without segmentation) are compared with the results prior to cleaning the signals acquired in an industrial setting.

	Before		After (no segmentation)		After (with segmentation)	
	MP	WP	MP	WP	MP	WP
Mean ± std	60 ± 50.7	36.2 ± 28.5	17.0 ± 18.0	16.2 ± 17.2	16.7 ± 16.0	16.5 ± 17.5
% Imp	-	-	71.7	55.2	72.2	54.4

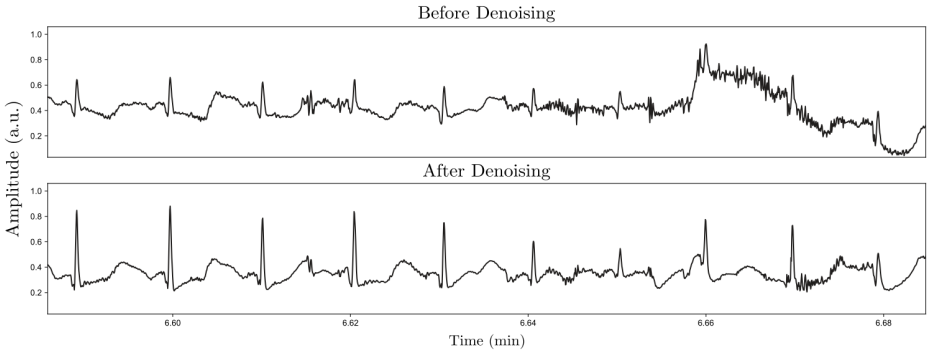


Figure 5.10: Example of the result obtained when applying the proposed model to a noisy input signal (shown in the upper plot) from the factory setting collection.

5.3.5 Discussion

Regarding the model's performance on different types of noise, the results from Table 5.4 show that the model is most and least effective in cleaning BW and EM, respectively, which is in accordance with results found in the literature. Unexpectedly, however, the mean results reveal no consistent relationship between the presence of more than one noise types and the model's performance, as it does not necessarily yield worse outcomes in these cases. For example, when denoising MA+BW noise, the RMSE is lower than when removing just MA noise and higher than when removing just BW noise. This can be interpreted as if the denoiser removes each type of noise independently and, for the same amount of SNR_{in}, when multiple noise sources are present, the error is around the mean of the obtained error for each type of noise alone. Overall, the model is able to efficiently remove a great part of the noise from the noisy ECG signals, achieving SNR_{imp} values of up to 30.6, 28.5, 26.5, and 24.2 dB on records with SNR_{in} of 0, 5, 7 and 10 dB, respectively.

Regarding employment of the BiGRU model to the data acquired in a real industrial setting, the results were satisfactory, for two main reasons. The first is the fact that the new data in which the model was tested differed greatly from the data with which it was trained: different environment (clinical setting versus factory), different protocol (steady versus in movement), and different electrode placement in the body (which alters the morphology of the signal). The second is that the workers were performing a variety of upper body work tasks (and some of them were very intense) which inherently results in a great amount of muscle activation noise summing to other noise sources from the workplace context (previously unseen by the model). The fact that the model was able to properly remove the main noise components of the signal (as it is visible in Fig. 5.9, for example) shows that it has a good generalization power and that it is not over-fitted neither for the PTB-XL ECG nor for the MIT-BIH NST noise sources.

Given that we are the first to use the PTB-XL database together with MIT-BIH Noise Stress database as basis for developing a denoiser to clean MA, EM and BW noise, there are no benchmarks available from the literature to which our results can be directly compared.

Table 5.6: Overview of key attributes of main related work.

Model	Parameters	ECG Database	Noise added	Real noisy ECG tests	Open-source
FFNN [157]	3003000 [1]	MIT-BIH Arrhythmia; MGH; QT [2]	EM [3]	Yes	Yes
DAE [121]	12220 [1]	PTB Diagnostic	GWN	No	No
CNN; LSTM [12]	8764758; 21266320 [1]	Synthetic data; MIT-BIH Arrhythmia	Synthetic random noise; EM [3]	No	No
LSTM [9]	9915352 [1]	Synthetic data; PTB Diagnostic	GWN	No	No
CDAE-LSTM [39]	10919980	MIT-BIH Arrhythmia	GWN	No	No
GAN [206]	281060	MIT-BIH Arrhythmia	EM, MA, BW [3]	No	No
DeepRTSNet [5]	346034	MIT-BIH Apnea	EM, MA, BW [3]	Yes	Yes
biGRU	26121	PTB-XL	EM, MA, BW [3]	Yes	Yes

For this reason, we opted for: (i) comparing our models to traditional denoising approaches and (ii) performing a qualitative comparison with state-of-the-art models. From Fig. 5.7, it is observable that our model's performance significantly surpasses widely used publicly available tools for ECG denoising. Regarding the state-of-the-art approaches reported in the literature and described in Section 5.3.2, we undertook a qualitative comparison, summarized in Table 5.6.

As already mentioned in the section 5.3.2, in most studies the models were trained using the MIT-BIH Arrhythmia database. Apart from this one the other databases that were used are: the MGH Waveform database, with data from 250 patients [209], the QT database, which included 105 recordings [98], the PTB Diagnostic, from 290 subjects [22], and the MIT-BIH Apnea, consisting of 18 records [75]. Compared to the size and inherent diversity of PTB-XL, the mentioned databases lack representative power. Regarding the noise addition step that is common to all approaches, limiting the training and testing of the model to the presence of gaussian white noise (as in [121], [12], and [39]) is very restrictive. In [157], the authors perform tests on one record from the MGH database, which contains real noisy signals (ie, there was not the need of artificially adding noise to it); however, the only paper that also performed tests on a non-clinical environment was [5]. Even so, the tests performed in the present study go a step further when the data collection is performed in a non-controlled environment, where the researchers had no interference on the tasks that were being performed.

In terms of computational complexity, our model has a parameter count of 2.6×10^4. Among the related works, from our estimates, the DAE from [121] has an even lower

parameter count; however, its training and testing were limited to Gaussian noise, significantly limiting its adaptability to other prevalent sources of noise. Our proposed model outperforms the remaining by at least a tenfold reduction in the number of parameters when compared to other studies. This advantage is critical, particularly for reducing testing time, especially on lightweight devices, where computational resources are constrained. It's worth noting that most of the existing models are not publicly accessible, rendering testing and utilization impossible. The FFNN model [157] and DeepRTSNet [5] are publicly accessible. While the former provides the code but not the trained model, the latter requires a paid subscription for access.

The primary limitation of this study pertains to the R-peak detection tool utilized, as it is not infallible. Nonetheless, we contend that the impact of this limitation is mitigated. Also, it's worth noting that the clean signals used for model training are not entirely devoid of noise (although the database is of very good quality); in the context of our study, the metrics employed in the initial phase compute the divergence between the provided output and the original signal. Consequently, an output that surpasses the original signal in terms of cleanliness might erroneously receive a lower performance score. Another limitation pertains to the absence of a ground-truth reference for evaluating the denoiser's performance on real-world noisy data. Nevertheless, it's worth noting that this challenge extends to the evaluation of most machine learning models, as real-world scenarios often lack definitive ground-truth references. Addressing this issue is imperative to effectively assess the model's generalization capabilities.

5.3.6 Key Takeaways

In this work, a lightweight GRU-based DL architecture is proposed to execute the task of ECG denoising. For the first time, a DL denoiser was trained and tested on the PTB-XL database for the task of removing MA, EM and BW noise from ECG signals. The developed approach cleans noisy ECG signals and outputs signals with good quality, while simultaneously preserving the trace of abnormal signals of patients. Furthermore, the model's robustness for application in real-world scenarios was tested on a large data collection that was performed in an industrial setting, during work. The results obtained from the new dataset emphasize the model's efficacy and generalization power, supporting the potential integration of the BiGRU denoiser into a comprehensive framework for monitoring the occupational health of workers through wearable data.

5.4 Noise Detection and Classification[2]

In the previous section, we presented a solution for removing the most typical noise types from ECG records. A more complete denoising pipeline should, however, include an ECG quality assessment tool, responsible for identifying and selecting segments of the signal requiring cleaning, ensuring that the denoiser is applied only where necessary.

Traditional methods for Signal Quality Assessment (SQA) for ECG signals typically rely on statistical, frequency or morphological features of the data. Statistical approaches, as the name suggests, are based on statistical metrics, such as variance and zero-crossing rate, which provide insights into the signal's variability, or kurtosis and skewness, which assess whether the signal's distribution falls within expected ranges [156, 87]. Rule-based approaches often combine multiple metrics to enhance the robustness of quality assessment; however, these methods depend on fixed thresholds, which may not perform consistently across different contexts or datasets. Frequency-based methods focus on analyzing the power spectrum of the signal. However, their effectiveness is significantly limited when the noise frequency band overlaps with the frequencies of the ECG signal, restricting their applicability. Morphology-based approaches, on the other hand, rely on clean ECG templates that are assumed to represent a noise-free signal, comparing new ECG data against these templates [156]. As discussed in Chapter 4, the shape of an ECG signal is influenced not only by electrode placement but also by the presence of CVD. Consequently, deviations from a 'typical' ECG pattern do not necessarily indicate the presence of noise, which poses challenges for the generalized use of morphology-based methods.

As argued both in Chapter 4 and in the previous section, DL models learn high-level features directly from ECG signals, allowing for more adaptive and scalable approaches compared to traditional approaches. The reviewed studies from the literature explore various DL approaches for ECG signal quality assessment and classification, primarily leveraging CNN and LSTM architectures. Zhou et al. [224] pioneered the use of 1D CNNs for binary classification of ECG quality, demonstrating their effectiveness over traditional methods. Building on this, Mondal [134] introduced a three-layer 1D CNN architecture incorporating first-order derivatives to enhance noise detection, while Liu et al. [110] proposed a dual-input model combining scalograms with handcrafted features. Extending into 2D representations, Huerta et al. [73], [74] applied transfer learning with image-based CNN models such as AlexNet and VGG16, achieving competitive results. Addressing temporal dependencies, J. Zhang et al. [219] employed LSTM structures combined with domain-specific features to enhance classification accuracy. To counteract dataset limitations, Zhou et al. [225] introduced a CGAN-based data augmentation technique, improving performance significantly. More recent approaches

[2]This section is based on the publication "Assessing Electrocardiogram Quality: A Deep Learning Framework For Noise Detection And Classification" by Márcia Monteiro, **Mariana Dias**, and Hugo Gamboa, published in the BIOSTEC 2025 Conference. In the present document, only a summary of the most relevant aspects of the study is provided; for a comprehensive overview, please refer to the open-access publication.

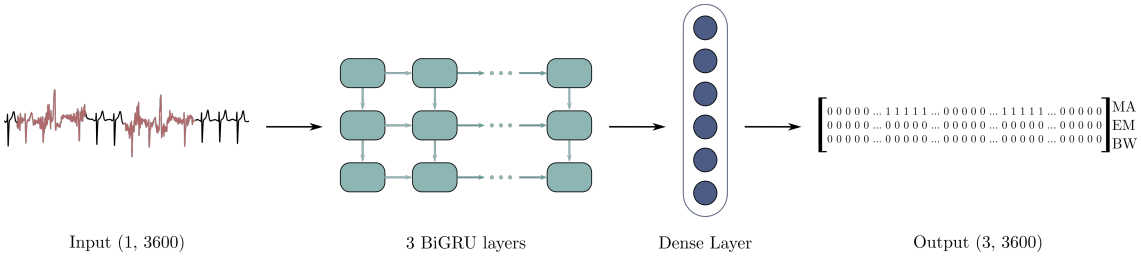


Figure 5.11: Schematic representation of the proposed model. The input is a signal with shape [1, 3600], where 3600 is the sequence length, and 1 represents the input size (number of features per time step) and the output is a tensor of shape [3, 3600], since each time step contains the multi label classification of the 3 noise classes.

integrate attention mechanisms, such as Jin et al.'s DAC-LSTM [83] and Zhong et al.'s DenseNet with Squeeze-and-Excitation modules [222], which improved feature selection and classification. Furthermore, transformer-based methods have emerged, with Chen et al. [32] proposing SwinDAE, leveraging 1D Swin Transformers to handle long recordings with reduced computational costs. Finally, X. Zhang et al. [220] explored a hybrid CNN-RNN model using residual recurrent modules (RRMs) to classify wearable ECG signals, achieving high accuracy but facing challenges in handling electrode motion artifacts.

The described methods are mostly limited to binary classification and do not provide additional details, such as the exact location or type of noise present. For this reason, a new approach is introduced in this section. Using a methodology very similar to the one outlined in Section 5.3 for training the noise removal model, a model for noise detection and classification was also developed. The data sources were, again, PTB-XL and MIT-BIH Noise Stress Test. The pre-processing steps and the methodology to create the noisy signals was the same as described in 5.3.3.1. The input to the model were, similarly, the corrupted 10-second signals and the main difference from the methods described in section 5.3.3.1 is that instead of producing a clean signal as output, the model generates a one-hot-encoded vector of the same length as the input signal, meaning that each timestep is classified as contaning, or not, each type of noise. Each dimension of the matrix corresponds to one of the three types of noise to be detected - MA, EM, and BW - where values of 0 and 1 indicate the absence or presence of noise, respectively, as represented in Figure 5.11.

The architecture includes three stacked GRU layers with bidirectional processing. A dropout layer follows each GRU layer to prevent overfitting by temporarily deactivating units across the feature space. The fully connected (FC) layer transforms the GRU output into a [batch_size, 3600, 3] tensor, classifying each time step into one of three states [MA, EM, BW]. The model outputs a sequence of vectors with dimensions [batch_size, 3600, 3], providing raw logit scores for each noise type at each time step. The loss function used for training was Binary Cross-Entropy with Logits Loss, which is well-suited for binary classification tasks where the model outputs raw logits for each class.

For performance evaluation, an individual confusion matrix was computed for each noise type to evaluate the model's ability to correctly predict whether each class (MA,

EM, BW) is `'Present'` or `'Absent'`. The matrix shows the counts of True Negatives (TN), False Positives (FP), False Negatives (FN) and True Positives (TP). Additionally, a general multi-label confusion matrix was calculated to evaluate the model's ability to detect each of the four categories: MA [1, _, _], EM [_, 1, _], BW [_, _, 1], and None [0, 0, 0]. This matrix summarizes the frequency of noise misclassification and helps to identify which classes are most commonly confused. It is important to note that *None* is not a distinct class but rather a result of no noise being present.

A grid search was performed (from the values presented in table 5.7) and the architecture of the model that lead to the lowest loss in the validation set has 3 bidirectional layers with a hidden size of 128, a dropout rate of 0.3, a learning rate of 0.001, and a batch size of 128.

The model performance was evaluated using accuracy, precision, recall, and F1-scores for three noise types. These results are summarized in Table 5.8. Accuracy was highest for EM noise at 92.86 %, followed by BW noise at 92.05 %, with the lowest accuracy for MA noise at 81.55 %. In precision, BW noise scored highest at 82.36 %, indicating fewer false positives, while EM and MA noise scored 79.35 % and 50.37 %, respectively. Recall was consistently high, with BW noise achieving 96.56 %, followed by MA at 90.14 % and EM at 85.26 %. The F1 score, balancing precision and recall, reflected these trends, with BW noise scoring 88.89 %, EM 82.19 %, and MA 64.62 %.

The model's performance metrics show that it performed best in detecting BW noise, achieving high precision and recall, while EM showed high accuracy but lower recall, indicating occasional missed detections. MA classification proved more challenging, with higher false positives and lower precision, though recall remained strong. The F1 score highlighted BW as the most effectively managed noise, with EM achieving a moderate balance and MA struggling with false positive and negative rates. Despite these challenges, the model demonstrated robustness in identifying noise presence across classes.

Table 5.7: Hyperparameter values explored during grid search

Hyperparameters	**Values**
Type of layers	GRU
Number of layers	3
Bidirectional	{ `True`, `False` }
Batch size	128
Hidden size	{ 64, 128, 256 }
Dropout rate	{ 0, 0.3, 0.5 }

Table 5.8: Performance Metrics

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

Its capability to simultaneously classify and localize noise types represents a significant advancement over existing Signal Quality Assessment (SQA) methods, offering a more comprehensive approach to ECG noise detection.

5.5 Final Remarks

This chapter presents advancements in ECG noise detection, classification and removal using lightweight DL methods. The proposed denoising model demonstrated robust performance in eliminating common noise types - MA, EM, and BW - while maintaining the integrity of the ECG signal. Notably, the model's performance remained consistent even when applied to real-world industrial data, highlighting its generalization capability beyond controlled environments. Similarly, the noise detection and classification model achieved competitive results, providing an automated and granular assessment of signal quality across diverse conditions. Together, these approaches contribute to enhancing ECG signal reliability, facilitating more accurate downstream analyses in both clinical and occupational settings.

Beyond the specific applications discussed, this work provides valuable insights into the broader field of DL for biosignal processing. The results underscore the importance of balancing complexity and computational efficiency, as the results showed - just like in Chapter 4 - that higher complexity does not necessarily mean enhanced performance. As biosignal acquisition moves outside clinical settings, solutions must prioritize adaptability to diverse and dynamic environments, ensuring robust performance across varied conditions.

While the noise removal and detection models were developed independently, the possibility of leveraging one for the other - such as reusing the denoiser in the detection model via transfer learning - emerged as a promising direction for future work. Such an approach could optimize resource utilization, accelerate convergence, and foster a more cohesive framework for biosignal analysis. This reflection sets the stage for the next chapter, which introduces the Neural Library for Biosignals, a framework aimed at developing modular and reusable DL tools for comprehensive biosignal processing.

In conclusion, the findings presented in this chapter contribute to the growing body of research demonstrating the effectiveness of DL in biosignal processing. They also pave the way for future explorations into unified, resource-efficient approaches that maximize the potential of DL in both research and practical applications.

6

The Neural Library

In Chapter 3, both in section 3.1.5 and 3.2.1, we analysed two complex and interconnected challenges of the field. On the more technical side, we discussed complexity, overfitting and generalization and, on a more ethical perspective, the environmental impact of Deep Learning (DL). Chapter 4 and Chapter 5 present two studies whose results show us that high complexity does not mean higher predictive power, reinforcing the shift toward more sustainable and efficient approaches. In this Chapter, we present `NeuralLib`, a DL-based framework for biosignal processing implemented upon three fundamental principles: modularity, efficiency, generalization.

6.1 Introduction and Objectives

Biosignals are time series that capture physiological processes or specific dimensions of those processes. As such, they are widely used for monitoring and extracting information about the body's functioning at a given moment or within a particular context. However, as discussed throughout this thesis, transforming raw signals into interpretable information is not straightforward - biosignals are non-stationary, noisy, and highly variable, both across individuals and within the same individual over time. DL has proven highly effective in addressing these challenges, as demonstrated in Chapter 4, where DL models extracted meaningful patterns from cardiovascular data, and Chapter 5, where they successfully identified and removed noise in biosignal recordings.

Meanwhile, growing awareness of the computational cost of DL models has raised concerns about the sustainability of current approaches, where most AI practitioners prioritize model efficacy, often drastically increasing complexity for only marginal gains in accuracy. Research has shown that it is possible to reduce resource usage while maintaining performance, particularly by leveraging hierarchical feature learning. Neural networks tend to learn general, transferable representations in their early layers, with deeper layers specializing in solving specific tasks. This makes reusing pre-trained models an efficient and practical approach, reducing the need for training from scratch. One way to achieve this is through Transfer Learning (TL) strategies, such as model-based TL, which allows

models trained on one task to be adapted to another with reduced additional training. Given the increasing adoption of DL in biosignal analysis, it is crucial to move toward a structured, resource-efficient approach that prioritizes model reusability and accessibility.

Standardizing model development can streamline the process of designing, training, and evaluating DL architectures, making it easier to compare approaches and optimize for efficiency. A structured framework should also support the organization and sharing of trained models, ensuring that researchers and practitioners can leverage existing knowledge instead of unnecessarily expending resources to develop similar models repeatedly. Also, allowing the adaptation of trained models for similar biosignal-related tasks prevents the need for new models to be trained from scratch for each new application. This not only minimizes energy consumption but also enables faster model deployment in real-world settings. Adopting modular design principles and efficient DL strategies makes it possible to enhance both the scalability and accessibility of DL applications in biosignal processing.

Within the scope of this thesis, the Neural Library was developed to embody these ideas, providing an open-source DL-based structured framework that promotes efficiency, reusability, scalability, and collaboration, empowering researchers and practitioners to apply state-of-the-art neural models to biosignal data. The primary objective of `NeuralLib` is to establish a modular, efficient, generalizable framework for biosignal processing using DL.

Specifically, `NeuralLib` aims to:

- Provide a systematic methodology for implementing DL models tailored to biosignal processing.
- Enable the training of new architectures from scratch while ensuring that pre-trained models are available for direct use.
- Promote transfer learning by offering tools to extract, recombine, and fine-tune components of trained models.
- Foster community-driven collaboration by making models and code publicly accessible for further development.
- Minimize computational overhead by encouraging efficient training and adaptation strategies.

`NeuralLib` is not intended to be a *one-size-fits-all* solution to every biosignal processing challenge. Instead, it aims to build a dynamic and expandable ecosystem of models, where components can be efficiently reconfigured to address diverse tasks. This vision aligns with the broader goal of generalization, as models are intended to be adapted to new domains and applications.

To support these objectives, the trained models and code repository are publicly available:

- Code repository: https://github.com/novabiosignals/NeuralLib
- Pre-trained models: https://huggingface.co/collections/novabiosignals/neurallib-deep-learning-models-for-biosignals- processing-6813ee129bc1bba8210b6948
- Code documentation: https://novabiosignals.github.io/NeuralLib-docs/

By fostering an open-source DL framework for biosignal analysis, NeuralLib seeks to simplify model deployment and foster collaboration within the scientific community while minimizing computational overhead through model sharing and reuse. This library is based on 3 principles: modularity, efficiency, and generalization. In the following section, these are contextualized and described.

All schematics in this chapter use a recurrent network structure as an illustrative example. This choice is purely for visualization purposes and does not imply that the described concepts are limited to recurrent architectures.

6.2 Principles of the Framework

6.2.1 Modularity

Modularity is a principle observed in various fields, from biology to computer science. It refers to the decomposition of a system into smaller, functionally independent components, known as modules. By breaking down complexity into manageable parts, modularity facilitates adaptability and enhances the overall efficiency of a system. In the context of computational models, this structured organization offers multiple advantages, such as improved scalability, reusability, and incremental development [191].

In DL, a module refers to any model or self-contained component within a model that performs a specific transformation. This transformation can be explicitly trained - such as in a standalone model designed for a particular task - or implicitly learned when the module consists of a layer or a set of layers within a larger model, as represented in Figure 6.1. In the latter case, the module's function is not predetermined but rather shaped by the training process, meaning its role may not correspond to an easily interpretable task [148]. Modularity in DL aligns with reusability, as it allows the reuse, repurpose, and recombination of components. It is also on the basis of the idea of generalizing to broader domains, ultimately leading to higher computational efficiency. The 3 principles discussed in the present section are, hence, inherently interconnected.

For leveraging pre-trained models and use knowledge transfer strategies, namely through model-based TL approaches (analysed in section 3.2.3), modular frameworks present some advantages, such as parameter efficiency. Instead of fine-tuning an entire model for a new task, modular approaches allow updating only relevant parts, making computation more efficient [148], as represented in figure 6.2. Moreover, it allows for progressive learning and adaptation, as new modules can be introduced over time, helping models learn continuously without catastrophic forgetting [148].

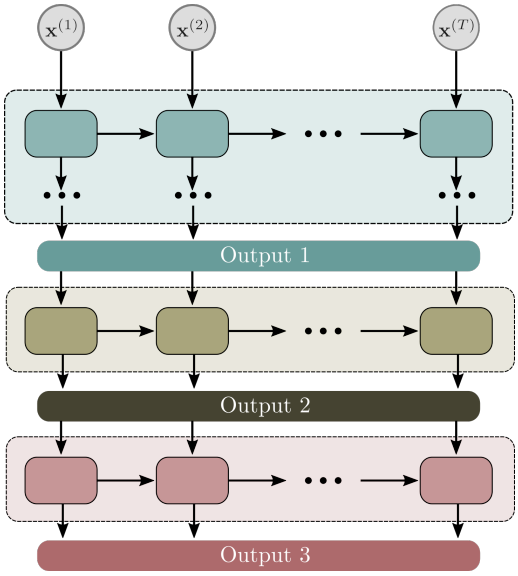


Figure 6.1: Schematic representation of `NeuralLib`'s modularity. The diagram depicts a network architecture composed of distinct modules, each represented by a different color. Each module performs a specific transformation - either as a single layer or a group of layers - and can be reused or adapted independently.

The AlphaPeptDeep framework is an example of the power of modularity in DL [217]. Designed for mass spectrometry-based proteomics, it employs a flexible architecture that allows researchers to predict peptide properties - such as retention time, ion mobility, and fragment intensities - using interchangeable DL models. Its modularity enables easy adaptation to new experimental conditions and peptide types by allowing users to refine and extend models without needing large datasets. This approach maximizes reusability by integrating pre-trained models and fine-tuning them for specific applications, drastically reducing the computational burden. Another example is the Interpretable Modular Deep Learning Framework for Video-Based Fall Detection illustrates modularity by breaking down fall detection into distinct, specialized components [42]. The system is composed of multiple independent modules that handle specific tasks. By allowing individual components to be activated or deactivated depending on privacy and monitoring requirements, the framework offers flexibility that makes it adaptable to different healthcare settings. Both of these studies effectively showcase how modular DL architectures enhance scalability, reusability, and flexibility.

Instead of models being designed in isolated, monolithic structures, modular approaches enable the community-driven sharing, expansion, and continuous refinement of models. It fosters more efficient AI development by reducing redundancy and encouraging the reuse of existing knowledge. As a result, modular DL contributes to a broader ecosystem where models and components can be interchanged, improved, and adapted to new challenges over time.

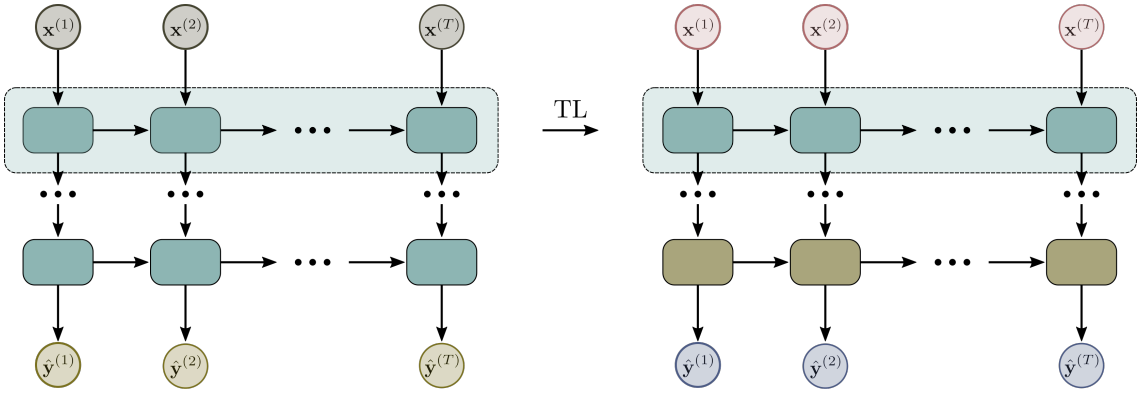


Figure 6.2: Schematic representation of modularity in Transfer Learning (TL), where pretrained parameters from the first layer (blue) of model A are transferred to model B. The different colors in the inputs x and outputs y represent different datasets, while the color differences in the layers indicate different parameter sets.

6.2.2 Efficiency

As discussed in section 3.2.1, the energy consumption required to train DL models is high due to several factors, including: (i) computationally intensive operations requiring high-performance GPUs, (ii) long training times, and (iii) the use of cooling systems in cases of high-energy consumption setups. Additionally, the manufacturing of hardware required for these computations contributes to environmental impact, both through resource extraction - associated with ecological disruption in mining areas - and the generation of electronic waste (e-waste) when devices become obsolete. To mitigate these issues, optimizing algorithms for computational efficiency is essential. In this regard, models with lower complexity offer, in general, a more sustainable alternative. Chapters 4 and 5 illustrate how less complex models can achieve competitive, state-of-the-art results, reinforcing their viability as efficient yet effective alternatives.

The principles of efficiency in DL parallel the well-known environmental sustainability framework of the three R's: Reduce, Reuse, and Recycle. First, reducing computational costs should be a priority, which can be achieved by designing smaller, more efficient models that require fewer resources to train and deploy. Next, reusing existing models before training new ones - leveraging pre-trained networks whenever possible - minimizes redundant computations and promotes knowledge sharing within the community. Finally, recycling models refers to the repurposing and fine-tuning of pre-trained architectures for new tasks - a process enabled by TL. This approach allows previously acquired knowledge to be adapted to different contexts, reducing the need for extensive training from scratch. By leveraging models that have already internalized generic patterns from large datasets, TL facilitates faster convergence, lowers data requirements, and significantly reduces computational overhead. In doing so, it reinforces both computational efficiency and the sustainable use of DL resources.

While DL research often emphasizes achieving the highest possible accuracy, this pursuit frequently comes at the expense of efficiency. It is common to find models claiming novelty based solely on a marginal increase in accuracy, yet these improvements often come with significantly higher computational costs [65], as discussed in 3.2.2. Moreover, variations in evaluation datasets, metric formulations, and testing conditions can make these small gains unreliable or context-dependent. Instead of focusing exclusively on performance benchmarks, a "good enough" approach prioritizes extracting meaningful and actionable insights while minimizing resource consumption. The goal is not simply to push error rates lower but to develop models that efficiently capture the essential patterns in data, balancing accuracy with sustainability. This shift in focus encourages more responsible AI development, ensuring that advancements in DL contribute meaningfully to both scientific progress truly and sustainable computing practices.

Efficiency is, thus, a ratio between performance and footprint. To measure the footprint, the metrics proposed in [127] for both training and inference are used: number of FLOPs (floating-point operations), number of parameters, and RAM consumption.

6.2.3 Generalization

Generalization in DL, as discussed in section 3.1.5 refers to a model's ability to perform well on previously unseen data, beyond the specific examples it was trained on. A well-generalized model can effectively capture the underlying patterns of the data rather than memorizing the training set, ensuring reliable performance in real-world scenarios.

While increasing model complexity can enhance representational power, it does not always translate to better generalization. In fact, as discussed in Chapters 4 and 5, as well as in the literature [65], excessive complexity can lead to overfitting - where the model performs well on training data but fails to adapt to new data. In contrast, simpler models with optimized architectures often achieve better generalization, particularly when trained on diverse and well-curated datasets. The work presented in Chapter 5 demonstrates that it is possible to generalize well to different contexts without the need for additional training or fine-tuning.

However, from the literature review presented in Section 3.1.5, it is known that in cases where test data does not follow the same distribution as the training data, generalization might be harder to achieve. Domain Generalization (DG) approaches are often employed to prevent performance drops in those cases, when data distribution shifts between training and deployment [223]. However, as DG aims to create models that generalize well across different data domains without using target domain data, it often requires more complex architectures and training strategies, increasing computational demands and making implementation challenging [103]. An alternative and more computationally efficient strategy is Domain Adaptation (DA), where models are adjusted to new data distributions using a limited amount of target domain data. Once more, TL plays a central role in this process, as pre-trained models can be fine-tuned for specific tasks with minimal

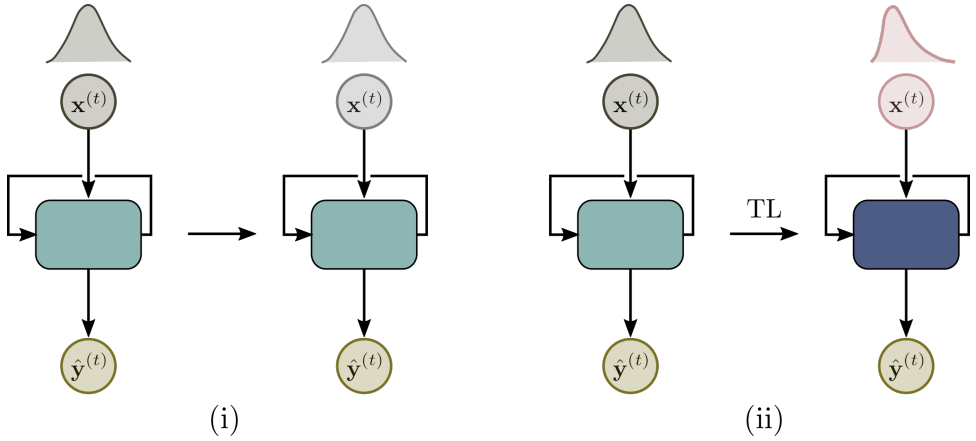


Figure 6.3: Schematic representation of two generalization scenarios: (i) the model successfully generalizes to a new dataset without requiring domain adaptation when the data distributions are sufficiently similar; (ii) when the new dataset has a different distribution, fine-tuning is necessary to achieve generalization.

labeled data from the new domain. This approach reduces the need for excessive model complexity, making it an effective alternative to domain generalization when adaptation data is available. For this reason, the present work prioritizes DA over DG, leveraging TL to achieve generalization across different data distributions in a computationally efficient manner.

Furthermore, modularity provides a structured approach to generalization by allowing models to systematically adapt to new tasks through the recombination and local updating of modules [148]. Instead of retraining entire models, modular architectures make it possible to transfer knowledge efficiently by integrating and fine-tuning only relevant components. This modular approach not only improves reusability but also enhances generalization while maintaining computational efficiency. The three principles are well aligned.

6.3 NeuralLib

The complexity of biosignal processing tasks, coupled with the growing availability of DL models, has created a need for flexible, scalable, and efficient frameworks that allow researchers and developers to streamline model development and deployment. `NeuralLib` was designed to address this need by providing a structured, modular framework that enables both training DL architectures from scratch and leveraging pre-trained models through TL.

`NeuralLib`'s core design principles - modularity, efficiency, and generalization - ensure that models can be easily adapted, reused, and shared. The framework includes a set of predefined biosignal architectures tailored for both regression and classification tasks, allowing users to experiment with various model configurations and training strategies. Additionally, it provides built-in functionalities for extracting, fine-tuning, and reusing

trained components, making it a practical tool for research and other applications.

The key elements of `NeuralLib` are illustrated in Figure 6.4. The framework adopts an object-oriented approach built around four core classes: `Architecture`, `ProductionModel`, `TLFactory`, and `TLModel`, each represented as a distinct block in the figure. The `Architecture` class provides the fundamental methods required to train models from scratch or retrain them from saved checkpoints. Once a model has been trained and tested, it can be uploaded to Hugging Face and wrapped as a `ProductionModel`, which stores not only the model itself but also essential metadata such as task type, performance metrics, and hyperparameters. To enable transfer learning, the `TLFactory` class offers mechanisms to extract and repurpose components from an existing `ProductionModel`, allowing the creation of a new `TLModel`. This model can then be fine-tuned, tested, and, if suitable, converted into a new `ProductionModel`, supporting iterative development and model reuse. The following subsections explain in more detail how these classes function and interact.

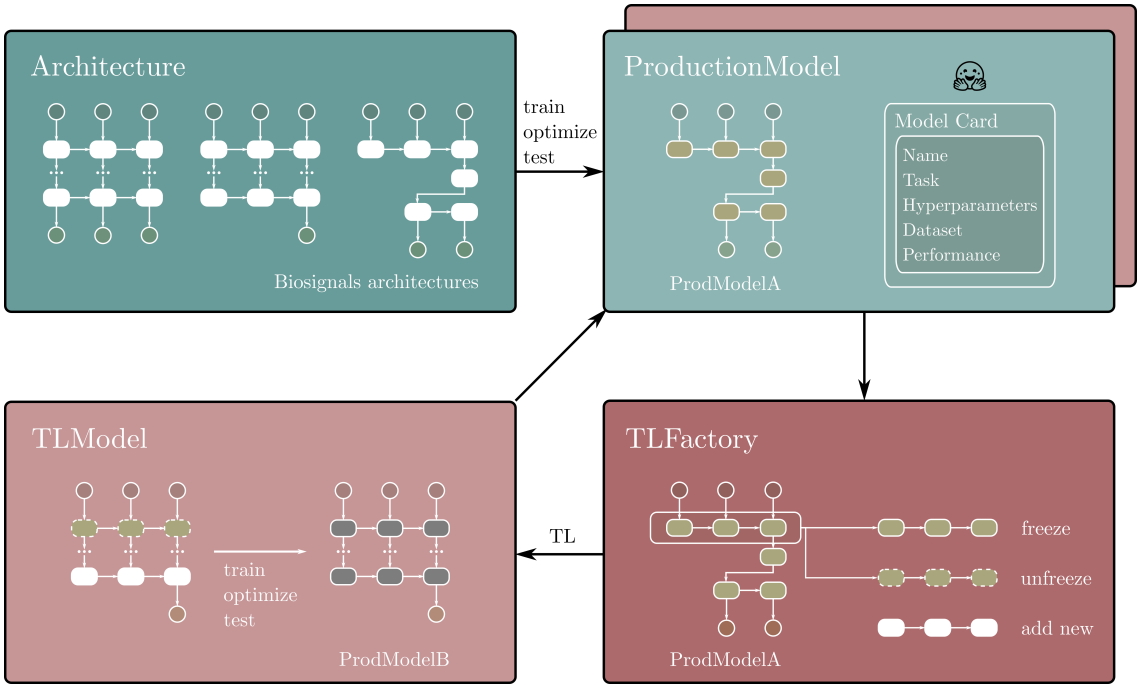


Figure 6.4: Schematic representation of the `NeuralLib` framework, illustrating the roles and interactions of its four core classes. Each class is represented in a block: `Architecture` for model development, `ProductionModel` for model packaging and sharing, `TLFactory` for Transfer Learning operations, and `TLModel` for models adapted from pre-trained parameters. Arrow directions illustrate the workflow. White nodes indicate untrained parameters; colored nodes indicate trained parameters, with each color representing a distinct set trained together.

6.3.1 Package structure

`NeuralLib` is organized in a way to facilitate development and extension. It was implemented in Python [200] using the Pytorch DL framework [147]. The framework is

structured into three sub-packages:

- `architectures/`: Contains the implementation of the main class `Architecture` and its methods, such as training and testing functions. Includes the implementation of several biosignals architectures designed for biosignals processing tasks. Also provides utilities for uploading trained and tested models to Hugging Face and share them with the community.
- `model_hub/`: Manages `ProductionModel` instances, allowing users to load trained models from Hugging Face, fine-tune, or extract trained components using `TLFactory` for creating and training `TLModel` objects.
- `utils/`: Contains auxiliary functions, including plotting tools, as well as dataset validation, handling, and loading, through the `DatasetSequence` class.

A detailed breakdown of the directory structure and its components is provided in Appendix A. The following subsections describe the two core subpackages of the library: `architectures` and `model_hub`.

6.3.2 Architectures

`architectures` contains the implementation of the `Architecture` class, providing a structured framework for defining, training, testing, and managing neural network models for biosignal processing. It integrates with PyTorch Lightning, streamlining the training workflow while providing checkpoint management and hyperparameter optimization via grid search. It also supports (optionally) the integration with TensorBoard for visualization of training metrics, model tracking, and logging.

6.3.2.1 Main Methods

The `architectures` subpackage provides a structured foundation for model development, ensuring a standardized approach to implementing and training neural networks, namely through the usage of the `Architecture` class and its methods. Among its key functionalities, the `train_from_scratch()` method enables the development of new models from scratch, handling dataset loading, model initialization, hyperparameter optimization (through grid search), and testing, while integrating mechanisms such as checkpointing and early stopping. The retrain method extends this functionality by allowing models to be further trained with additional data or longer training durations while preserving previous training history. To facilitate evaluation, `test_on_test_set()` applies the trained model to the provided test set, ensuring that generalization performance is systematically assessed, while `test_on_single_signal()` allows for an in-depth analysis of model behavior on individual inputs. These testing functions allow for the addition of a post processing step to be applied to the output of the network, if `post_process_fn()` is provided. Additionally, the class provides utilities for managing training metadata and

checkpointing, such as `save_training_information()`, which records essential details of the training process for reproducibility and tracking. Finally, the `upload_to_hugging()` method allows a trained and tested model to be uploaded to Hugging Face. This process uploads four files: a `.json` file containing training metadata, a `.yaml` file with the model's hyperparameters, a `.pth` file storing the model's learned parameters - these three are automatically generated during training - and a `README.md` file, created at the time of upload to serve as the model card. These methods collectively enable `NeuralLib` to support efficient model development, tracking, and reuse, aligning with the overarching principles of modularity and efficiency in DL.

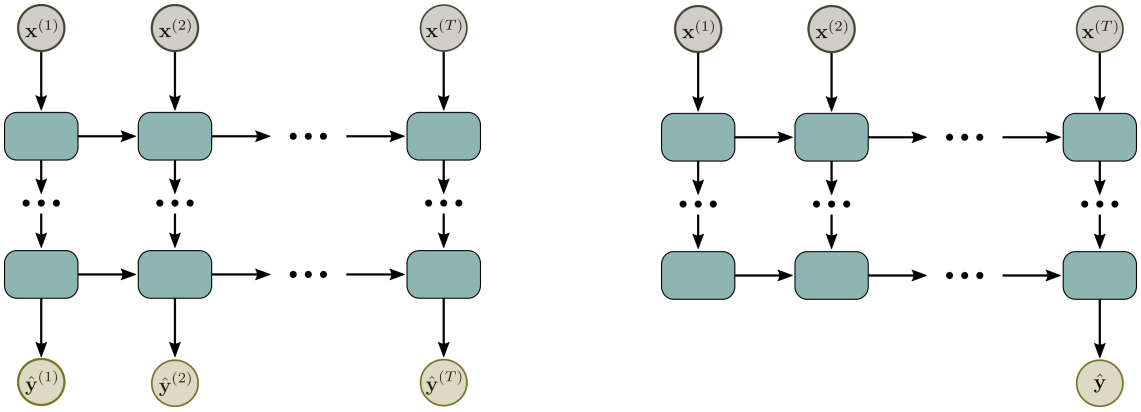
In Appendix B, it is possible to find a full hands-on guide on the different functionalities of the library. Sections B.1 and B.2 encompass functionalities from the `architectures` subpackage. The development of any model entails selecting an appropriate architecture paradigm, defining the task type, and specifying the input and output shapes. The following section outlines the reasoning behind the design of the biosignal processing architectures included in the library.

6.3.2.2 Architectures for Biosignals Processing

Biosignals processing encompasses a wide range of tasks, such as regression, classification, or signal generation, each demanding tailored computational approaches depending on the nature of the signals and the desired outputs. The selection of a neural network architecture is driven by the specific problem being addressed. These tasks are categorized here based on the structure of the output, as the input is consistently a time series - specifically, a biosignal. Three main types are considered:

- **sequence-to-sequence:** each time step in the input corresponds to a transformed value at the same time step in the output, illustrated in Figure 6.5 (a)
- **sequence-to-one:** the entire input sequence is mapped to a single value or vector (not a timeseries), shown in Figure 6.5 (b)
- **encoder-decoder:** the input sequence is transformed into an output sequence that does not align one-to-one with the input in terms of time steps and includes an intermediate step of compressing the input into a latent representation, represented in Figure 6.6.

While encoder-decoder models are formally a subclass of sequence-to-sequence architectures, they are treated here as a distinct category based on output structure, as described above. These three architecture categories, collectively referred to as *biosignals architectures*, are represented in the illustration of `NeuralLib` in the `Architecture` block of Figure 6.4. Each model implementation inherits from the base `Architecture` class, allowing consistent training workflows.



(a) (b)

Figure 6.5: Schematic representation of (a) sequence-to-sequence and (b) sequence-to-one architectures.

Additionally, the nature of the task itself may involve multiclass or multilabel classification, or regression. Binary classification is implemented as multi-label with 1 class. All biosignals architectures inherit from `Architecture` class. The combination of these dimensions - input-output mapping and task type - guided the design choices for biosignal-specific architectures. Given the temporal dependencies inherent to biosignals, two DL paradigms were selected:

- **GRU:** As discussed in Chapter 3, biosignals exhibit complex temporal dependencies, non-stationarity, and high inter- and intra-subject variability, requiring models capable of capturing both short-term and long-term dependencies. GRUs are well-suited for biosignal processing due to their ability to efficiently model temporal patterns while mitigating the vanishing gradient problem, making them particularly effective for sequence-to-sequence and sequence-to-one tasks with limited computational overhead.

- **Transformers:** Transformers, with their self-attention mechanisms, excel at modeling long-range dependencies and capturing global relationships across time, making them highly effective for biosignals with complex temporal structures or when large amounts of training data are available [10].

The complementary strengths of these architectures provide a versatile foundation for `NeuralLib`, ensuring adaptability across different biosignal processing tasks. *Biosignals architectures* include GRU- and Transformer-based models implemented in the three formats described above: sequence-to-sequence, sequence-to-one, and encoder-decoder. Each of these implementations supports different task types - namely, multiclass or multilabel classification, or regression - depending on the configuration selected by the user. Table 6.1 summarizes the loss functions used during training, along with the corresponding output shapes for each model category and task type.

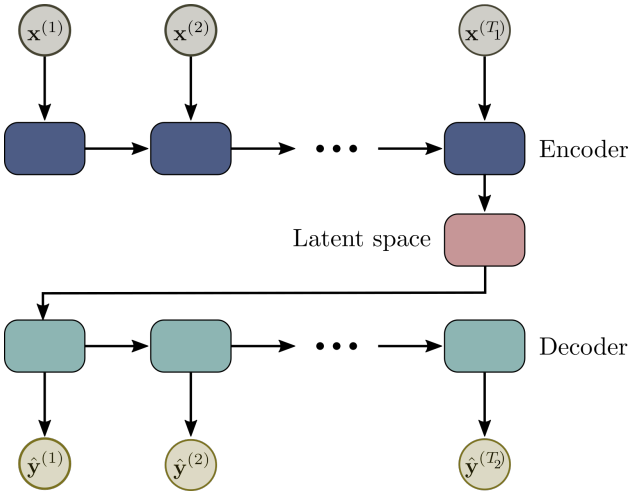


Figure 6.6: Schematic representation of encoder-decoder architectures. Unlike the sequence-to-sequence architecture, the input and ouput timeseries have different lenghts, T_1 and T_2, respectively.

Table 6.1: Loss functions used during model training and corresponding output shapes, according to the task type and model architecture. *n_classes* denotes the number of classes, *len* is the output sequence length, and *n_feat* refers to the number of channels or features in the input sequence. BCE stands for Binary Cross Entropy, and MSE stands for Mean Squared Error.

Task		**Loss Function**	**Y shape**	
			Seq2one	**Seq2seq / ED**
Classification	**Multi-label**	*BCEwithLogits*	$(1, n\_classes)$	$(len, n\_classes)$
	Multiclass	*CrossEntropy*	$()$	(len)
Regression		*MSE*	$(1, n\_feat)$	(len, n_feat)

6.3.3 Model Hub

The `model_hub` subpackage provides streamlined tools for managing and reusing pre-trained models within `NeuralLib`. It enables users to efficiently deploy trained models, perform inference, and apply transfer learning techniques to adapt existing models to new biosignal tasks. Specifically, it allows users to:

- Load pre-trained models from Hugging Face
- Perform inference on biosignal data using `ProductionModel` instances
- Extract and inject pre-trained weights into new `TLModel` instances for TL
- Freeze/unfreeze layers to optimize training using `TLFactory`.

6.3.3.1 Production Models

Once a model is trained, tuned, and tested, it can be exported and uploaded to Hugging Face for public access and reuse. The `ProductionModel` class, part of the `model_hub`, is designed to facilitate the import and use of these models for both inference and parameter

extraction. It inherits from the `Architecture` class and extends its functionality with automatic loading of weights, training metadata, and model configuration directly from Hugging Face repositories. The `list_production_models()` function displays all models available in *NeuralLib*'s Hugging Face repository. Additionally, models from other repositories can be loaded by explicitly providing the full Hugging Face repository ID.

6.3.3.2 Transfer Learning Factory

The `TLFactory` class enables the adaptation of existing models to new tasks or datasets using model-based TL. Through this class, it is possible to:

- Load an existing `ProductionModel`.
- Extract selected layers (e.g., GRU layers) from the original model.
- Inject these layers into a new architecture via `TLModel`.
- Configure which layers are frozen (to preserve learned features) and which are trainable (to adapt to the new task).

The resulting `TLModel` is a new architecture that leverages pre-trained components and is ready for fine-tuning. Since `TLModel` inherits from `Architecture`, it retains full compatibility with training, testing, and deployment methods, maintaining a consistent model definition across the framework.

6.4 Final Remarks

This chapter introduced `NeuralLib`, a structured DL framework developed to address core challenges in biosignal processing. Built upon the foundational principles of modularity, efficiency, and generalization, `NeuralLib` aims to simplify the development and reuse of neural models, offering a consistent and flexible environment for training, evaluation, and deployment.

The design of the library emphasizes scalability, enabling users to either build models from scratch or adapt pre-trained ones through TL. Features such as integration with Hugging Face for model sharing and support for layer-wise reuse and fine-tuning via `TLFactory` position `NeuralLib` within the broader transition toward unified and more sustainable practices in DL. While progress has been made in improving model performance, the challenge that remains is to achieve these advances with greater computational efficiency.

`NeuralLib` was conceptualized upon the experimental findings from earlier chapters. For instance, the results from both Chapters 4 and 5 demonstrated that simpler architectures can perform competitively in real-world tasks, and these findings helped shape the library's core philosophy, which prioritizes computational efficiency without compromising performance.

Although `NeuralLib` currently supports a set of task-specific architectures and workflows, its modular structure allows for easy extension to other architecture paradigms. By encouraging model sharing and reuse, it contributes to the democratization of DL for biosignals, fostering more transparent, reproducible, and sustainable practices.

In this sense, `NeuralLib` serves not only as a practical tool but also as a step toward more efficient and collaborative research in biomedical signal analysis.

7

Conclusions

This chapter revisits the research objectives outlined at the beginning of the thesis, discussing how each was addressed and the key findings that emerged from this work. The discussion is framed around the four main objectives (O1-O4) and the research questions (RQ1-RQ2), highlighting contributions, limitations, and potential paths for future research.

7.1 Main Conclusions

This thesis explores the potential of Deep Learning (DL) in biosignal processing, with two primary use cases: cardiovascular disease classification and ECG noise removal. Based on the findings from these studies and on state-of-the-art literature - with the ultimate goal of contributing to more unified and sustainable solutions for biosignal challenges using DL - a systematic framework was proposed. Each objective tackled a distinct aspect of this problem space, from optimizing model complexity to promoting real-world applicability and supporting open, reproducible research. Next, we revisit these objectives and discuss the key findings related to each.

O1: Apply DL techniques for cardiovascular disease (CVD) classification, analysing model complexity and performance.

Electrocardiography can be very effective in the early identification of CVD. For instance, in diseases that affect the heart's electrical conduction, the diagnosis is directly done through the recognition of abnormal waveforms. DL facilitates ECG analysis by enabling the direct use of raw signals and automating processing steps that would otherwise require some level of expertise to be carried out.

While most literature on the application of DL methods to ECG relies on the 1D signal, 2D representations have also been successfully employed, as they highlight anomalies in ways that differ from the raw 1D format. Acknowledging that distinct representations can convey complementary information, we investigated whether their combined use could

enhance performance, and, if so, how this integration would influence the complexity of solving the classification task.

This objective was primarily addressed in Chapter 4, where distinct DL models were applied to classify CVDs using ECG data. The investigation focused on understanding how different data formats and model architectures impact classification performance. To achieve this, we evaluated the use of 2D Convolutional Neural Networks (CNNs) for processing image representations of ECG signals, and both 1D CNNs and Recurrent Neural Networks (RNNs) - including Gated Recurrent Unit (GRU) and Long Short Term Memory (LSTM) networks - for processing the original 1D signals. CNNs were explored for their ability to extract spatial features , while RNN-based architectures were assessed for their capability to model temporal dependencies in biosignals. A multimodal approach, combining 1D CNNs for feature extraction with RNNs for sequential learning, was also tested. Since each format entails specific architectural choices, in parallel to assessing model performance in each case, we reflected on complexity variation, using, for that, the number of trained parameters for each approach.

The results demonstrated that using models with higher capacity did not improve classification performance in the present challenge. While improvements were found in classification sensitivity with the multimodal approach, which is important in disease diagnosis, the raw ECG signal processed by a unidirectional 3-layer GRU yielded the overall best performance, with a model size under 250 thousand parameters. Competitive results were achieved while maintaining efficiency, reinforcing the importance of balancing representational power and resource constraints.

O2: Develop a DL model for ECG noise removal and assess its generalization across diverse datasets.

Ensuring data quality and readability is crucial for effective biosignal analysis, yet this becomes particularly challenging when it is acquired in uncontrolled environments. To address this objective, Chapter 5 introduced a GRU-based deep learning model designed to remove noise from ECG signals. The model was first trained and evaluated using real ECGs merged with noise recordings to simulate contamination with various types of noise. Subsequently, it was tested on real-world data acquired in an industrial setting with multiple noise sources.

The results demonstrated that the model effectively improved signal quality while preserving key morphological features. Quantitative metrics such as signal-to-noise ratio (SNR) and mean squared error (MSE) confirmed the model's ability to denoise signals across a range of conditions. Despite being trained only on synthetically contaminated data, the model maintained robust performance when applied to real signals, suggesting a strong capacity for generalization across acquisition contexts and device types. Notably, this generalization was achieved on a BiGRU model with 26 thousand parameters, comparatively small in relation to state-of-the-art approaches.

In addition to denoising, a similar model extended this work to perform automatic detection and classification of noise types, enabling a more nuanced understanding of signal contamination. This capacity for both signal enhancement and noise diagnosis supports its applicability in real-world scenarios where biosignals are subject to multiple sources of degradation.

Overall, the findings from this objective highlight the potential of DL to support robust and generalizable preprocessing pipelines, especially in challenging environments where traditional noise reduction techniques fall short.

O3: Develop a systematic DL framework that supports modularity and efficiency in biosignal processing tasks.

Chapter 6 addressed this objective through the development of `NeuralLib`, a structured DL framework designed to promote modularity, reusability, and efficiency in biosignal processing. The library integrates standardized training and testing routines, support for model sharing via Hugging Face, and tools for TL through layer-wise weight reuse and fine-tuning. These features facilitate the adaptation of trained models to new tasks or datasets without the need to retrain from scratch, supporting more sustainable and accessible practices in DL.

Efficiency in deep learning is influenced by several factors, including the type of hardware used, the duration and complexity of training, and the cost of inference. While inference remains an unavoidable step, the energy required for training can be significantly reduced, when leveraging pre-trained models. Fine-tuning these models offers a middle ground by reducing training time and computational resources, provided that the new task is sufficiently related to the original one. However, in cases where data distributions or task definitions differ substantially, fine-tuning may lead to catastrophic forgetting and hinder model performance.

By offering a flexible structure that accommodates model reuse and selective retraining, `NeuralLib` contributes to minimizing redundant computational efforts and supports the broader shift toward more efficient and environmentally conscious DL practices in biosignal analysis.

O4: Ensure accessibility and scalability by making the models, training methodology, and codebase open-source for the research community.

Transparency, reproducibility, and accessibility are critical pillars for advancing scientific research. In response to these principles, Objective 4 focused on ensuring that all developed resources - models, training methodologies, and the underlying codebase - were made publicly available. The result is `NeuralLib`, an open-source deep learning framework designed to facilitate the development, evaluation, and reuse of neural models in biosignal processing.

To promote accessibility, `NeuralLib` was made freely available on GitHub, accompanied by detailed documentation and tutorials (see Appendix B) to support adoption by the broader research community. In parallel, pre-trained models developed during the thesis were shared through the Hugging Face platform, making them readily accessible for inference or fine-tuning across diverse applications. These models are accompanied by training metadata and model cards to ensure proper understanding of their use cases and limitations.

By encouraging model sharing and reusability, this objective supports scalability beyond the scope of this work, enabling future researchers to build on existing resources without the need to retrain models from scratch. It also reduces the environmental and computational cost typically associated with training deep learning models, reinforcing the commitment to more sustainable AI practices.

Overall, this objective consolidates the idea that scientific contributions should be not only technically robust but also open, reusable, and extensible - contributing to a collaborative ecosystem where knowledge is continuously refined and shared.

7.1.1 Reflections on the Research Questions

RQ1: How can DL be effectively applied to biosignal processing tasks?

This thesis explored the application of deep learning in biosignal processing through two distinct case studies: cardiovascular disease classification and ECG noise removal. These examples demonstrate the versatility of DL in handling real-world biosignals, which are often noisy, variable, and context-dependent. By adapting model architectures to task-specific requirements, this work showed how careful design can significantly enhance model effectiveness.

Additionally, the choice of input representation was found to influence model outcomes, with 1D, 2D, and multimodal formats each emphasizing different aspects of the signal. Importantly, findings across both studies confirmed that simpler architectures can perform competitively, supporting a more efficient and sustainable approach to DL applications in this field.

Generalization across datasets and acquisition conditions was also a key focus, particularly in the denoising task, where the model trained on mixed-source data maintained robust performance in a real industrial scenario. Finally, the development of `NeuralLib` provided a structured, extensible framework that supports training, evaluation, and reuse of models across biosignal tasks, illustrating not just how DL can be applied effectively - but also how it can be deployed efficiently and reproducibly in practice.

RQ2: What strategies can be implemented to improve computational efficiency in DL models while maintaining performance in biosignal applications?

The findings of this thesis highlight several complementary strategies that can improve computational efficiency in DL models for biosignal applications, without compromising performance. First, both case studies presented in Chapters 4 and 5 showed that relatively simple architectures - such as GRUs with a limited number of layers and hidden units - were capable of achieving competitive results across diverse tasks. These findings reinforce the idea that lower complexity does not necessarily come at the cost of accuracy.

Transfer learning emerged as a key strategy for minimizing training costs. By reusing parts of models trained on related tasks and adapting them to new datasets, the cost of training can be significantly reduced - and in some cases, avoided entirely if inference on a pre-trained model is sufficient. The importance of task similarity was also underscored, since reusing models in very different domains may lead to performance degradation due to catastrophic forgetting. To support these strategies, `NeuralLib` was developed as a modular and extensible framework that facilitates model reuse, sharing, and adaptation. The framework includes support for measuring efficiency using objective metrics - such as FLOPs, number of parameters, and memory consumption - ensuring that performance improvements can be considered in balance with resource demands.

Overall, this work advocates for a shift in focus from maximizing accuracy at all costs toward prioritizing sustainable and efficient DL practices, particularly relevant in biosignal processing contexts where computational resources are often limited.

7.1.2 Limitations and Other Considerations

While this work presents promising approaches and contributions to DL for biosignal processing, its limitations should be acknowledged to contextualize the findings and guide future developments.

The first study presented in this thesis (Chapter 4) is primarily limited by the fact that models were not evaluated on ECG signals from datasets outside the PTB-XL database. This restriction was due to the absence of other public databases containing the same diagnostic labels. As a result, the generalization capacity of the models across acquisition settings and populations could not be fully assessed.

In the second study (Chapter 5), additional limitations arise from the nature of the data and the evaluation procedures. Although the training data was constructed using real ECG and noise recordings, the clean ECGs used as references were not entirely noise-free. This could introduce bias in the evaluation, particularly in cases where a denoised output improves upon the reference signal but receives a lower score. Moreover, no ground-truth was available for the real-world noisy recordings, limiting the ability to rigorously assess the model's real-world performance. This issue, while common in biosignal research, highlights the ongoing challenge of evaluating models under operational conditions.

While the relationship between model complexity and performance was explored, no generalizable rule could be established. As observed in both case studies, comparatively light models were often able to achieve competitive results. However, performance varied depending on the task and data characteristics, suggesting that complexity-performance trade-offs must be assessed on a case-by-case basis rather than through universal assumptions.

Additionally, only two use cases were explored in depth: cardiovascular disease classification and ECG noise removal. Although both tasks are highly relevant and complementary in nature, they do not represent the full range of applications encountered in biosignal research. Further work is needed to explore more tasks and more modalities other than ECG, in order to validate the generalizability of the conclusions.

Despite the emphasis on designing more sustainable deep learning workflows - discussed in Chapter 3 and reflected in the design of `NeuralLib` (Chapter 6) - accurately quantifying the environmental footprint of each model was beyond the scope of this thesis. Energy consumption depends not only on the architecture itself but also on numerous external factors including hardware type, power source, and cooling systems. Additionally, while practices such as grid search help optimize performance, they involve training multiple models, which may offset the environmental gains made through model simplification or efficient inference.

Finally, although TL is a core concept in this thesis, particularly in the `NeuralLib` framework, it was not empirically tested across the two case studies. Given that both studies used ECG data, applying cross-task transfer could have yielded valuable insights into the potential for reuse and adaptation in biosignal contexts.

7.1.3 Perspectives for Future Research

This work opens several avenues for continued exploration. One of the most promising is the deeper investigation into the potential and limitations of TL in biosignal processing. While TL is widely recognized as effective when source and target tasks are sufficiently aligned, the boundaries of this alignment remain underexplored. Future work could focus on systematically assessing the impact of task similarity, data domain, and signal modality on TL performance. Key questions include: How similar do the source and target domains need to be for TL to remain beneficial? Is there a measurable threshold beyond which performance degrades? How does model complexity influence TL success? These questions are especially relevant in light of evidence suggesting that domain generalization benefits from increased architectural complexity - raising the question of whether the same applies to domain adaptation.

In parallel, continued development of NeuralLib presents valuable opportunities. While the current framework supports modularity, model reuse, and pretraining workflows, future updates could expand architecture diversity (e.g., including convolutional RNNs, attention-based mechanisms), extend support to other signal types (e.g., EMG,

EEG, ACC), and incorporate model compression or quantization tools for edge-device deployment. Another aspect involves evaluating alternative hyperparameter optimization strategies beyond grid search, which may provide comparable tuning quality at a lower computational cost. Additionally, future versions of the framework could explore strategies to improve efficiency at inference time, such as compression techniques (e.g., pruning and quantization) or learning strategies like knowledge distillation, which allow smaller models to retain the performance of larger ones while significantly reducing computational demands.

Finally, future research could explore human-in-the-loop and active learning methods to combine expert input with DL models in biosignal labeling or noise identification, reducing reliance on large, fully labeled datasets while maintaining high performance.

7.2 Contributions

Throughout my PhD, I have had the opportunity to contribute to open science, teaching, and mentorship.

7.2.1 Research Contributions

Along with the team from LibPhys-Biosignals, I have been actively involved in large data collection efforts as part of the Operator - MIT Portugal Project, conducting two extensive studies:

- A controlled laboratory experiment with 37 subjects, aimed at identifying physiological markers for fatigue detection.
- A field study in industrial settings with 46 participants, assessing occupational risk on an automobile assembly line.

Within this project, I also took part in exploring the potential of Respiratory Inductive Plethysmography (RIP) and Heart Rate-related metrics for tracking fatigue and cardiorespiratory response in assembly line work, studies that lead to the publication of two journal papers and three conference proceedings. Additionally, I contributed to the PrevOccupAI project, by participating in the development of a portal for occupational health assessment through questionnaires.

My international research experience was strengthened through the Fulbright Program, where I spent four months at MIT. During this time, I collaborated with MIT.nano's Immersion Lab, implementing code for automatic motion capture marker labeling.

Beyond technical contributions, I contributed to the LibPhys-Biosignals group by conceptualizing and drafting a research proposal for a new project to advance the lab's work in biosignal processing and DL applied to Occupational Health.

7.2.2 Teaching and Mentorship

My PhD journey also included teaching responsibilities at NOVA School of Science and Technology (FCT NOVA), where I was an invited teaching assistant in the Physics Department, covering electrophysiology and thermodynamics practical classes. Additionally, I was invited to teach a guest lecture on Deep Learning theory in the Machine Learning course, as well as two workshops on biosignal processing.

I have also had the opportunity to contribute in the supervision and guidance of three master's students, supporting their academic development in biosignal processing and Machine Learning.

7.2.3 Open Science and Scientific Dissemination

As part of my commitment to open science, I contributed to the development of `NeuralLib`, a software for biosignal processing, which will be made freely available to the research community, supporting reproducibility and facilitating future studies in the field.

The developed research has been shared with the scientific community through three oral presentations at conferences and three poster presentations, contributing to discussions on biosignal processing, fatigue detection, and DL for physiological signal analysis.

7.3 List of Publications

7.3.1 Journal Papers

- Dias, M., Silva, L., Folgado, D., Nunes, M. L., Cepeda, C., Cheetham, M., & Gamboa, H. (2023). Cardiovascular load assessment in the workplace: A systematic review. International Journal of Industrial Ergonomics, 96. https://doi.org/10.1016/j.ergon.2023.103476

- Dias, M., Probst, P., Silva, L. & Gamboa, H. Cleaning ECG with Deep Learning: A Denoiser Tested in Industrial Settings. SN COMPUT. SCI. 5, 699 (2024). https://doi.org/10.1007/s42979-024-03017-7

- Narotamo, H., Dias, M., Santos, R., Carreiro, A. V., Gamboa, H., & Silveira, M. (2024). Deep learning for ECG classification: A comparative study of 1D and 2D representations and multimodal fusion approaches. Biomedical Signal Processing and Control, 93, 106141. https://doi.org/10.1016/j.bspc.2024.106141

- Silva L, Dias M, Folgado D, Nunes M, Namburi P, Anthony B, Carvalho D, Carvalho M, Edelman E, & Gamboa, H. Respiratory Inductance Plethysmography to Assess Fatigability during Repetitive Work. Sensors. 2022; 22(11):4247. https://doi.org/10.3390/s22114247

- Furk, D., Silva, L., Dias, M., Fujão, C., Probst, P., Liu, H., & Gamboa, H. (2024). Cardiorespiratory Response to Workload Volume and Ergonomic Risk: Automotive

Assembly Line Operators' Adaptations. Applied Sciences, 14(9), 3921. https://doi.org/10.1016/j.ergon.2023.103476

7.3.2 Conference Papers

- Dias, M., Probst, P., Silva, L., Gamboa, H. (2023). Cleaning ECG with Deep Learning: A Denoiser Based on Gated Recurrent Units. In: Camarinha-Matos, L.M., Ferrada, F. (eds) Technological Innovation for Connected Cyber Physical Spaces. DoCEIS 2023. IFIP Advances in Information and Communication Technology, vol 678. Springer, Cham. https://doi.org/10.1007/978-3-031-36007-7_11

- Carvalho, D., Silva, L., Carvalho, M., Dias, M., Costa, N., Folgado, D., Nunes, M., Gamboa, H., Andza, K., Edelman, E. (2023). Cardiovascular Reactivity (CVR) During Repetitive Work in the Presence of Fatigue. Intelligent Human Systems Integration (IHSI 2023): Integrating People and Intelligent Systems. AHFE (2023) International Conference. AHFE Open Access, vol 69. AHFE International, USA. http://doi.org/10.54941/ahfe1002833

- Carvalho, D., Silva, L., Carvalho, M., Dias, M., Costa, N., Folgado, D., Lua, M., Gamboa, H., Edelman, E. (2023). Heart rate variability during repetitive work in the presence of fatigue. In: Francisco Rebelo and Zihao Wang (eds) Ergonomics In Design. AHFE (2023) International Conference. AHFE Open Access, vol 77. AHFE International, USA. http://doi.org/10.54941/ahfe1003433

- Furk, D., Silva, L., Dias, M., Probst, P., & Gamboa, H. (2024). Cardiorespiratory Adaptations to Work Volume on an Automobile Assembly Line. In BIOSTEC (1) (pp. 71-81). https://doi.org/10.5220/0012587800003657

- Monteiro, M., Dias, M., & Gamboa, H. (2025). Assessing Electrocardiogram Quality: A Deep Learning Framework For Noise Detection And Classification. In Proceedings of the 18th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2025) - Volume 1, pages 793-804. https://doi.org/10.5220/0013313800003911

Bibliography

[1] S. Abbaspourazad et al. "Large-scale Training of Foundation Models for Wearable Biosignals". In: (2023-12). URL: http://arxiv.org/abs/2312.05409 (cit. on pp. 43, 44).

[2] H. Abbass. "Editorial: What is Artificial Intelligence?" In: *IEEE Transactions on Artificial Intelligence* 2 (2 2021-04), pp. 94–95. ISSN: 26914581. DOI: 10.1109/TAI.2021.3096243 (cit. on p. 20).

[3] L. A. Abdulla and M. S. Al-Ani. "A review study for electrocardiogram signal classification". In: *UHD Journal of Science and Technology* 4.1 (2020), pp. 103–117 (cit. on pp. 46, 47).

[4] J. M. Adams. "The value of worker well-being". In: *Public Health Reports* 134.6 (2019), pp. 583–586 (cit. on p. 67).

[5] P. Aghaomidi et al. "DeepRTSNet: Deep Robust Two-Stage Networks for ECG Denoising in Practical Use Case". In: *IEEE Access* 10 (2022), pp. 128232–128249. DOI: 10.1109/ACCESS.2022.3225899 (cit. on pp. 83, 84).

[6] I. Ahmad et al. "Robust Epileptic Seizure Detection Based on Biomedical Signals Using an Advanced Multi-View Deep Feature Learning Approach". In: *IEEE journal of biomedical and health informatics* 28 (10 2024). ISSN: 2168-2208. DOI: 10.1109/JBHI.2024.3396130. URL: https://pubmed.ncbi.nlm.nih.gov/38696293/ (cit. on p. 41).

[7] Z. Ahmad et al. "ECG heartbeat classification using multimodal fusion". In: *IEEE Access* 9 (2021), pp. 100615–100626 (cit. on pp. 48–51, 54, 60, 63).

[8] M. Al-Bouwarthan et al. "A Field Evaluation of Construction Workers' Activity, Hydration Status, and Heat Strain in the Extreme Summer Heat of Saudi Arabia". In: *Annals of Work Exposures and Health* 64 (5 2020-06), pp. 522–535. ISSN: 2398-7308. DOI: 10.1093/ANNWEH/WXAA029. URL: https://academic.oup.com/annweh/article/64/5/522/5812589 (cit. on p. 17).

[9] K. Antczak. "Deep Recurrent Neural Networks for ECG Signal Denoising". In: (2018-07). DOI: 10.48550/arxiv.1807.11551. URL: https://arxiv.org/abs/1807.11551v3 (cit. on pp. 70, 71, 83).

[10] A. Anwar et al. "Transformers in biosignal analysis: A review". In: *Information Fusion* 114 (2025-02), p. 102697. ISSN: 1566-2535. DOI: 10.1016/J.INFFUS.2024.102697 (cit. on pp. 3, 30, 42, 99).

[11] Z. Arman et al. "Physiological workload evaluation by means of heart rate monitoring during motor-manual clearcutting operations". In: *International Journal of Forest Engineering* 32 (2 2021-05), pp. 91–102. ISSN: 1494-2119. DOI: 10.1080/14942119.2021.1868238. URL: https://www.tandfonline.com/doi/abs/10.1080/14942119.2021.1868238 (cit. on p. 17).

[12] C. T. Arsene, R. Hankins, and H. Yin. "Deep learning models for denoising ECG signals". In: *European Signal Processing Conference* 2019-September (2019-09). ISSN: 22195491. DOI: 10.23919/EUSIPCO.2019.8902833 (cit. on pp. 70, 83).

[13] G. I. Ash et al. "Establishing a Global Standard for Wearable Devices in Sport and Exercise Medicine: Perspectives from Academic and Industry Stakeholders". In: *Sports Medicine* 51 (11 2021-11), pp. 2237–2250. ISSN: 11792035. DOI: 10.1007/S40279-021-01543-5/TABLES/4. URL: https://link.springer.com/article/10.1007/s40279-021-01543-5 (cit. on p. 2).

[14] D. Bahdanau, K. Cho, and Y. Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014) (cit. on p. 55).

[15] T. Baltrusaitis, C. Ahuja, and L.-P. Morency. "Multimodal Machine Learning: A Survey and Taxonomy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2 (2019-02), 423–443. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2018.2798607. URL: https://doi.org/10.1109/TPAMI.2018.2798607 (cit. on pp. 48, 49).

[16] M. A. Barbosa et al. "Evaluation of the physical workload in coffee production: biomechanical and physiological aspects". In: *Coffee Science* 10 (1 2015), pp. 83–90 (cit. on p. 17).

[17] Y. Bengio, P. Simard, and P. Frasconi. "Learning Long-Term Dependencies with Gradient Descent is Difficult". In: *IEEE Transactions on Neural Networks* 5 (2 1994), pp. 157–166. ISSN: 19410093. DOI: 10.1109/72.279181 (cit. on p. 26).

[18] A. Berthelot et al. "Estimating the environmental impact of Generative-AI services using an LCA-based methodology". In: *Procedia CIRP* 122 (2024-01), pp. 707–712. ISSN: 2212-8271. DOI: 10.1016/J.PROCIR.2024.01.098 (cit. on p. 34).

[19] K. van der Bijl, M. Elgendi, and C. Menon. "Automatic ECG quality assessment techniques: A systematic review". In: *Diagnostics* 12.11 (2022), p. 2578 (cit. on p. 76).

[20] A. Bizzego, G. Gabrieli, and G. Esposito. "Deep Neural Networks and Transfer Learning on a Multivariate Physiological Signal Dataset". In: *Bioengineering 2021, Vol. 8, Page 35* 8 (3 2021-03), p. 35. ISSN: 2306-5354. DOI: 10.3390/BIOENGINEERING8030035. URL: https://www.mdpi.com/2306-5354/8/3/35/htmhttps://www.mdpi.com/2306-5354/8/3/35 (cit. on pp. 39, 41).

[21] M. Bolpagni et al. "Personalized Stress Detection Using Biosignals from Wearables: A Scoping Review". In: *Sensors 2024, Vol. 24, Page 3221* 24 (10 2024-05), p. 3221. ISSN: 1424-8220. DOI: 10.3390/S24103221. URL: https://www.mdpi.com/1424-8220/24/10/3221/htmhttps://www.mdpi.com/1424-8220/24/10/3221 (cit. on pp. 1, 13).

[22] R. Bousseljot, D. Kreiseler, and A. Schnabel. "Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet". In: (1995) (cit. on p. 83).

[23] L. Bouza, A. Bugeau, and L. Lannelongue. "How to estimate carbon footprint when training deep learning models? A guide and review". In: *Environmental Research Communications* 5 (11 2023-11), p. 115014. ISSN: 2515-7620. DOI: 10.1088/2515-7620/ACF81B. URL: https://iopscience.iop.org/article/10.1088/2515-7620/acf81bhttps://iopscience.iop.org/article/10.1088/2515-7620/acf81b/meta (cit. on pp. 33, 35).

[24] N. T. Bui and G. S. Byun. "The comparison features of ECG signal with different sampling frequencies and filter methods for real-time measurement". In: *Symmetry* 13 (8 2021-08). ISSN: 20738994. DOI: 10.3390/sym13081461 (cit. on p. 51).

[25] D. Bustos et al. "Applicability of Physiological Monitoring Systems within Occupational Groups: A Systematic Review". In: *Sensors (Basel, Switzerland)* 21 (21 2021-11). ISSN: 1424-8220. DOI: 10.3390/S21217249. URL: https://pubmed.ncbi.nlm.nih.gov/34770556/ (cit. on pp. 14, 18).

[26] L. Cai, J. Gao, and D. Zhao. "A review of the application of deep learning in medical image classification and segmentation". In: *Annals of translational medicine* 8.11 (2020) (cit. on p. 53).

[27] *Cardiovascular diseases*. World Health Organization (WHO). 2021. URL: https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1 (visited on 2022-08-23) (cit. on p. 14).

[28] *Cardiovascular diseases*. World Health Organization (WHO). 2023. URL: https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1 (cit. on p. 46).

[29] D. Carvalho et al. "Cardiovascular Reactivity (CVR) During Repetitive Work in the Presence of Fatigue". In: *Intelligent Human Systems Integration (IHSI 2023): Integrating People and Intelligent Systems* 69.69 (2023). DOI: 10.54941/ahfe1002833 (cit. on p. 76).

[30] D. Carvalho et al. "Heart rate variability during repetitive work in the presence of fatigue". In: *Ergonomics In Design*. 14th International Conference on Applied Human Factors and Ergonomics (AHFE 2023). AHFE International, 2023 (cit. on p. 76).

[31] S. Chatterjee et al. "Review of noise removal techniques in ECG signals". In: *IET Signal Processing* 14 (9 2020-12), pp. 569–590. ISSN: 1751-9683. DOI: 10.1049/IET-SPR.2020.0104 (cit. on pp. 69, 73).

[32] G. Chen et al. "SwinDAE: Electrocardiogram Quality Assessment Using 1D Swin Transformer and Denoising AutoEncoder". In: *IEEE journal of biomedical and health informatics* 27 (12 2023-12), pp. 5779–5790. ISSN: 2168-2208. DOI: 10.1109/JBHI.2023.3314698. URL: https://pubmed.ncbi.nlm.nih.gov/37698969/ (cit. on p. 86).

[33] L. Cheng et al. "Robotic arm control system based on brain-muscle mixed signals". In: *Biomedical Signal Processing and Control* 77 (2022-08), p. 103754. ISSN: 1746-8094. DOI: 10.1016/J.BSPC.2022.103754 (cit. on p. 13).

[34] F. Chollet. *Deep learning with Python*. Simon and Schuster, 2021 (cit. on pp. 20, 31).

[35] F. Chollet. "On the Measure of Intelligence". In: *CoRR* abs/1911.01547 (2019). arXiv: 1911.01547. URL: http://arxiv.org/abs/1911.01547 (cit. on p. 31).

[36] J. Chung et al. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In: (2014-12). URL: https://arxiv.org/abs/1412.3555v1 (cit. on pp. 27, 28).

[37] J. Chung et al. "Gated Feedback Recurrent Neural Networks". In: *32nd International Conference on Machine Learning, ICML 2015* 3 (2015-02), pp. 2067–2075. URL: https://arxiv.org/abs/1502.02367v4 (cit. on p. 27).

[38] S. Collins. "Occupational Factors, Fatigue, and Cardiovascular Disease". In: *Cardiopulmonary Physical Therapy Journal* 20 (2 2009-06), p. 28. ISSN: 2374-8907. URL: /pmc/articles/PMC2845267//pmc/articles/PMC2845267/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC2845267/ (cit. on p. 13).

[39] E. Dasan and I. Panneerselvam. "A novel dimensionality reduction approach for ECG signal via convolutional denoising autoencoder with LSTM". In: *Biomedical Signal Processing and Control* 63 (2021-01), p. 102225. ISSN: 1746-8094. DOI: 10.1016/J.BSPC.2020.102225 (cit. on pp. 70, 83).

[40] H. Dehghan et al. "Combination of wet bulb globe temperature and heart rate in hot climatic conditions: The practical guidance for a better estimation of the heat strain". In: *International Journal of Environmental Health Engineering* | 1 (2 2012). DOI: 10.4103/2277-9183.96006. URL: http://www.ijehe.org (cit. on p. 17).

[41] A. Diker et al. "A novel application based on spectrogram and convolutional neural network for ECG classification". In: *2019 1st International Informatics and Software Engineering Conference (UBMYK)*. IEEE. 2019, pp. 1–6 (cit. on pp. 48, 49, 54).

[42] M. Dutt et al. "An Interpretable Modular Deep Learning Framework for Video-Based Fall Detection". In: *Applied Sciences 2024, Vol. 14, Page 4722* 14 (11 2024-05), p. 4722. ISSN: 2076-3417. DOI: 10.3390/APP14114722. URL: https://www.mdpi.com/2076-3417/14/11/4722/htmhttps://www.mdpi.com/2076-3417/14/11/4722 (cit. on p. 92).

[43] S. Eom, S. Eom, and P. Washington. "SIM-CNN: Self-supervised Individualized Multimodal Learning for Stress Prediction on Nurses Using Biosignals". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 14315 LNCS (2024), pp. 155–171. ISSN: 1611-3349. DOI: 10.1007/978-3-031-47679-2_12. URL: https://link.springer.com/chapter/10.1007/978-3-031-47679-2_12 (cit. on p. 42).

[44] Eurostat. *Hours of work - Annual statistics*. 2022. URL: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Hours_of_work_-_annual_statistics#In_2020_more_people_worked_fewer_hours_than_usual (visited on 2022-08-23) (cit. on p. 13).

[45] A. Farahani et al. "A Brief Review of Domain Adaptation". In: (2021), pp. 877–894. ISSN: 2569-7080. DOI: 10.1007/978-3-030-71704-9_65. URL: https://link.springer.com/chapter/10.1007/978-3-030-71704-9_65 (cit. on p. 32).

[46] H. I. Fawaz et al. "Transfer learning for time series classification". In: *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018* (2018-07), pp. 1367–1376. DOI: 10.1109/BIGDATA.2018.8621990 (cit. on p. 39).

[47] G. Franchini. "GreenNAS: A Green Approach to the Hyperparameters Tuning in Deep Learning". In: *Mathematics 2024, Vol. 12, Page 850* 12 (6 2024-03), p. 850. ISSN: 2227-7390. DOI: 10.3390/MATH12060850. URL: https://www.mdpi.com/2227-7390/12/6/850/htmhttps://www.mdpi.com/2227-7390/12/6/850 (cit. on p. 35).

[48] D. Fuller et al. "Reliability and Validity of Commercially Available Wearable Devices for Measuring Steps, Energy Expenditure, and Heart Rate: Systematic Review". In: *JMIR Mhealth Uhealth 2020;8(9):e18694 https://mhealth.jmir.org/2020/9/e18694* 8 (9 2020-09), e18694. ISSN: 22915222. DOI: 10.2196/18694. URL: https://mhealth.jmir.org/2020/9/e18694 (cit. on p. 2).

[49] M. C. Garcia and T. Vieira. "Surface electromyography: Why, when and how to use it". In: *Revista andaluza de medicina del deporte* 4.1 (2011), pp. 17–28 (cit. on p. 8).

[50] A. S. George, A. S. H. George, and A. S. G. Martin. "The Environmental Impact of AI: A Case Study of Water Consumption by Chat GPT". In: *Partners Universal International Innovation Journal* 1 (2 2023-04), pp. 97–104. ISSN: 2583-9675. DOI: 10.5281/ZENODO.7855594. URL: https://puiij.com/index.php/research/article/view/39 (cit. on p. 33).

[51] A. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2019 (cit. on p. 22).

[52] A. Goldberger. *Goldberger's clinical electrocardiography*. Vol. 10. Elsevier, 2013 (cit. on pp. 8, 45, 46).

[53] A. L. Goldberger et al. "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals". In: *circulation* 101.23 (2000), e215–e220 (cit. on pp. 68, 71).

[54] P. González-Zamora, V. H. Benitez, and J. Pacheco. "A feature-based processing framework for real-time implementation of muscle fatigue measurement". In: *Cluster Computing* 26 (1 2023-02), pp. 385–394. ISSN: 15737543. DOI: 10.1007/S10586-021-03437-7/TABLES/5. URL: https://link.springer.com/article/10.1007/s10586-021-03437-7 (cit. on p. 14).

[55] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016 (cit. on pp. 20–22, 26).

[56] M. Goswami et al. "MOMENT: A Family of Open Time-series Foundation Models". In: *Proceedings of Machine Learning Research* 235 (2024-02), pp. 16115–16152. ISSN: 26403498. URL: https://arxiv.org/abs/2402.03885v3 (cit. on pp. 43, 44).

[57] A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. Vol. 385. Springer Berlin Heidelberg, 2012. ISBN: 978-3-642-24796-5. DOI: 10.1007/978-3-642-24797-2. URL: https://link.springer.com/10.1007/978-3-642-24797-2 (cit. on pp. 25–27).

[58] K. Greff et al. "LSTM: A Search Space Odyssey". In: *IEEE Transactions on Neural Networks and Learning Systems* 28 (10 2017), pp. 2222–2232. DOI: 10.1109/TNNLS.2016.2582924 (cit. on p. 26).

[59] M. Guhdar, R. J. Mstafa, and A. O. Mohammed. "Hybrid Deep Learning Model for epileptic seizure classification by using 1D-CNN with multi-head attention mechanism". In: (2025-01). URL: https://arxiv.org/abs/2501.10342v1 (cit. on p. 41).

[60] B. Guignard et al. "Validity, reliability and accuracy of inertial measurement units (IMUs) to measure angles: application in swimming". In: *Sports Biomechanics* (2024-10). ISSN: 17526116. DOI: 10.1080/14763141.2021.1945136. URL: https://www.tandfonline.com/doi/abs/10.1080/14763141.2021.1945136 (cit. on p. 9).

[61] D. Gupta et al. "Review of ECG arrhythmia classification using deep neural network". In: *Materials Today: Proceedings* (2021) (cit. on p. 46).

[62] J. E. Hall and A. C. Guyton. *Textbook of Medical Physiology*. 12th. Vol. XXXIII.2. Saunders Elsevier, 2011. ISBN: 978-1-4160-4574-8 (cit. on p. 9).

[63] M. Hammad, Y. Liu, and K. Wang. "Multimodal biometric authentication systems using convolution neural network based on different level fusion of ECG and fingerprint". In: *IEEE Access* 7 (2018), pp. 26527–26542 (cit. on p. 50).

[64] H. Han et al. "Multimodal multi-instance learning for long-term ECG classification". In: *Knowledge-Based Systems* 270 (2023), p. 110555. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2023.110555. URL: https://www.sciencedirect.com/science/article/pii/S0950705123003052 (cit. on p. 50).

[65] N. U. Haq et al. "Computationally efficient deep learning models for diabetic retinopathy detection: a systematic literature review". In: *Artificial Intelligence Review 2024 57:11* 57 (11 2024-09), pp. 1–44. ISSN: 1573-7462. DOI: 10.1007/S10462-024-10942-9. URL: https://link.springer.com/article/10.1007/s10462-024-10942-9 (cit. on pp. 32, 36, 94).

[66] S. Hochreiter. "The vanishing gradient problem during learning recurrent neural nets and problem solutions". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02 (1998), pp. 107–116 (cit. on p. 26).

[67] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9 (8 1997-11), pp. 1735–1780. ISSN: 08997667. DOI: 10.1162/NECO.1997.9.8.1735 (cit. on p. 26).

[68] G. Holste et al. "Efficient deep learning-based automated diagnosis from echocardiography with contrastive self-supervised learning". In: *Communications Medicine 2024 4:1* 4 (1 2024-07), pp. 1–10. ISSN: 2730-664X. DOI: 10.1038/s43856-024-00538-3. URL: https://www.nature.com/articles/s43856-024-00538-3 (cit. on p. 36).

[69] A. Holtermann et al. "The health paradox of occupational and leisure-time physical activity". In: *British journal of sports medicine* 46 (4 2012-03), pp. 291–295. ISSN: 1473-0480. DOI: 10.1136/BJSM.2010.079582. URL: https://pubmed.ncbi.nlm.nih.gov/21459873/ (cit. on p. 18).

[70] J. J. Hopfield. "Artificial Neural Networks". In: *IEEE Circuits and Devices Magazine* 4 (5 1988), pp. 3–10. ISSN: 87553996. DOI: 10.1109/101.8118 (cit. on p. 21).

[71] J. Huang et al. "ECG arrhythmia classification using STFT-based spectrogram and convolutional neural network". In: *IEEE access* 7 (2019), pp. 92871–92880 (cit. on pp. 48, 49).

[72] S.-C. Huang et al. "Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines". In: *NPJ digital medicine* 3.1 (2020), pp. 1–9 (cit. on pp. 50, 51, 55).

[73] A. Huerta et al. "Quality assessment of very long-term ECG recordings using a convolutional neural network". In: *2019 7th E-Health and Bioengineering Conference, EHB 2019* (2019-11). DOI: 10.1109/EHB47216.2019.8970077 (cit. on p. 85).

[74] A. Huerta et al. "Comparison of Pre-Trained Deep Learning Algorithms for Quality Assessment of Electrocardiographic Recordings". In: (2020-12), pp. 1–4. DOI: 10.1109/EHB50910.2020.9280217 (cit. on p. 85).

[75] Y. Ichimaru and G. Moody. "Development of the polysomnographic database on CD-ROM". In: *Psychiatry and clinical neurosciences* 53.2 (1999), pp. 175–177 (cit. on p. 83).

[76] J. L. Id, A. Kozak, and A. Nienhaus. "Prevalence and occupational risk factors of musculoskeletal diseases and pain among dental professionals in Western countries: A systematic literature review and meta-analysis". In: *PLOS ONE* 13 (12 2018). DOI: 10.1371/journal.pone.0208628. URL: https://doi.org/10.1371/journal.pone.0208628 (cit. on p. 14).

[77] *Identification and recognition of occupational diseases: Criteria for incorporating diseases in the ILO list of occupational diseases. Meeting of Experts on the Revision of the List of Occupational Diseases (Recommendation No. 194)*. International Labour Organization (ILO). 2009-10. ISBN: 978-92-2-122823-3 (cit. on p. 13).

[78] M. Iman, H. R. Arabnia, and K. Rasheed. "A Review of Deep Transfer Learning and Recent Advancements". In: *Technologies 2023, Vol. 11, Page 40* 11 (2 2023-03), p. 40. ISSN: 2227-7080. DOI: 10.3390/TECHNOLOGIES11020040. URL: https://www.mdpi.com/2227-7080/11/2/40/htmhttps://www.mdpi.com/2227-7080/11/2/40 (cit. on pp. 37, 38).

[79] S. M. Iqbal et al. "Advances in healthcare wearable devices". In: *npj Flexible Electronics 2021 5:1* 5 (1 2021-04), pp. 1–14. ISSN: 2397-4621. DOI: 10.1038/s41528-021-00107-x. URL: https://www.nature.com/articles/s41528-021-00107-x (cit. on p. 12).

[80] J. Jeong. "EEG dynamics in patients with Alzheimer's disease". In: *Clinical Neurophysiology* 115 (7 2004-07), pp. 1490–1505. ISSN: 1388-2457. DOI: 10.1016/J.CLINPH.2004.01.001 (cit. on p. 1).

[81] A. R. Jha. *Mastering PyTorch: Create and deploy deep learning models from CNNs to multimodal models, LLMs, and beyond*. Packt Publishing Ltd, 2024 (cit. on p. 21).

[82] Z. Jia et al. "Personalized Meta-Federated Learning for IoT-Enabled Health Monitoring". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2024). ISSN: 19374151. DOI: 10.1109/TCAD.2024.3388908 (cit. on p. 9).

[83] Y. Jin et al. "A novel attentional deep neural network-based assessment method for ECG quality". In: *Biomedical Signal Processing and Control* 79 (2023-01), p. 104064. ISSN: 1746-8094. DOI: 10.1016/J.BSPC.2022.104064 (cit. on p. 86).

[84] L. Johari et al. "Deep Learning-Based Noise Reduction Techniques in Electronic Signal Processing". In: *Proceedings of 9th International Conference on Science, Technology, Engineering and Mathematics: The Role of Emerging Technologies in Digital Transformation, ICONSTEM 2024* (2024). DOI: 10.1109/ICONSTEM60960.2024.10568634 (cit. on p. 40).

[85] A. M. Jones and H. Carter. "The Effect of Endurance Training on Parameters of Aerobic Fitness". In: *Sports Med* 29 (2000), pp. 373–386 (cit. on p. 16).

[86] S. L. Joshi, R. A. Vatti, and R. V. Tornekar. "A survey on ECG signal denoisingtechniques". In: *Proceedings - 2013 International Conference on Communication Systems and Network Technologies, CSNT 2013* (2013), pp. 60–64. DOI: 10.1109/CSNT.2013.22 (cit. on pp. 67, 68).

[87] C. Karmakar et al. "Robustness of electrocardiogram signal quality indices". In: (2022). DOI: 10.1098/rsif.2022.0012 (cit. on pp. 75, 85).

[88] N. Ketkar. *Deep Learning with Python: A Hands-on Introduction*. 1st ed. Apress, 2017. ISBN: 978-1-4842-2766-4. DOI: 10.1007/978-1-4842-2766-4 (cit. on pp. 22, 23, 27, 31).

[89] M. G. Khan. *Rapid ECG interpretation*. Ed. by C. P. Cannon. New Jersey: Springer, 2008 (cit. on p. 8).

[90] H. Kim et al. "SleepWatcher: Detecting sleep apnea/hypopnea syndrome from wearable devices using deep learning". In: *Biomedical Signal Processing and Control* 103 (2025-05), p. 107460. ISSN: 1746-8094. DOI: 10.1016/J.BSPC.2024.107460 (cit. on p. 41).

[91] D. P. Kingma and J. L. Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2014-12). URL: https://arxiv.org/abs/1412.6980v9 (cit. on p. 23).

[92] M. Kivimäki and I. Kawachi. "Work Stress as a Risk Factor for Cardiovascular Disease". In: *Current cardiology reports* 17 (9 2015-09). ISSN: 1534-3170. DOI: 10.1007/S11886-015-0630-8. URL: https://pubmed.ncbi.nlm.nih.gov/26238744/ (cit. on p. 14).

[93] A. Klemetti et al. "Systematic Literature Review on Cost-Efficient Deep Learning". In: *IEEE Access* 11 (2023), pp. 90158–90180. ISSN: 21693536. DOI: 10.1109/ACCESS.2023.3275431 (cit. on p. 36).

[94] P. Kligfield et al. "Recommendations for the Standardization and Interpretation of the Electrocardiogram". In: *Circulation* 115 (10 2007-03), pp. 1306–1324. ISSN: 00097322. DOI: 10.1161/CIRCULATIONAHA.106.180200. URL: https://www.ahajournals.org/doi/10.1161/CIRCULATIONAHA.106.180200 (cit. on p. 68).

[95] N. Krause, R. J. Brand, and G. A. Kaplan. "Occupational physical activity, energy expenditure and 11-year progression of carotid atherosclerosis". In: *Scandinavian Journal of Work Environment & Health* 33 (6 2007), pp. 405–424. ISSN: 0355-3140 (cit. on p. 14).

[96] A. K. Kumar et al. "Deep learning for predicting respiratory rate from biosignals". In: *Computers in Biology and Medicine* 144 (2022-05), p. 105338. ISSN: 0010-4825. DOI: 10.1016/J.COMPBIOMED.2022.105338 (cit. on p. 40).

[97] S. Kwon et al. "At-home wireless sleep monitoring patches for the clinical assessment of sleep quality and sleep apnea". In: *Science advances* 9 (21 2023-05). ISSN: 2375-2548. DOI: 10.1126/SCIADV.ADG9671. URL: https://pubmed.ncbi.nlm.nih.gov/37224243/ (cit. on p. 42).

[98] P. Laguna et al. "A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG". In: *Computers in cardiology 1997*. IEEE. 1997, pp. 673–676 (cit. on p. 83).

[99] J. Laitala et al. "Robust ECG R-peak Detection Using LSTM". In: vol. 20. 2020, pp. 1104–1111. ISBN: 9781450368667. DOI: 10.1145/3341105.3373945. URL: https://doi.org/10.1145/3341105.3373945 (cit. on p. 46).

[100] G. Lee and S. Lee. "Deep-Learning Domain Adaptation to Improve Generalizability across Subjects and Contexts in Detecting Construction Workers' Stress from Biosignals". In: *Journal of Computing in Civil Engineering* 38 (3 2024-02), p. 04024010. ISSN: 0887-3801. DOI: 10.1061/JCCEE5.CPENG-5665. URL: https://ascelibrary.org/doi/abs/10.1061/JCCEE5.CPENG-5665https://ascelibrary.org/doi/10.1061/JCCEE5.CPENG-5665 (cit. on p. 40).

[101] Y. J. Lee et al. "Artificial intelligence on biomedical signals: technologies, applications, and future directions". In: *Med-X 2024 2:1* 2 (1 2024-12), pp. 1–30. ISSN: 2731-8710. DOI: 10.1007/S44258-024-00043-1. URL: https://link.springer.com/article/10.1007/s44258-024-00043-1 (cit. on pp. 3, 29, 41).

[102] S. Legg and M. Hutter. "A Collection of Definitions of Intelligence". In: *Frontiers in Artificial Intelligence and Applications* 157 (2007-06), pp. 17–24. ISSN: 18798314. URL: https://arxiv.org/abs/0706.3639v1 (cit. on p. 20).

[103] B. Li et al. "Why Robust Generalization in Deep Learning is Difficult: Perspective of Expressive Power". In: *Advances in Neural Information Processing Systems* 35 (2022-05). ISSN: 10495258. URL: https://arxiv.org/abs/2205.13863v3 (cit. on pp. 32, 94).

[104] H. Z. Li and P. Boulanger. "A Survey of Heart Anomaly Detection Using Ambulatory Electrocardiogram (ECG)". In: *Sensors (Basel, Switzerland)* 20 (5 2020-03). ISSN: 14248220. DOI: 10.3390/S20051461 (cit. on pp. 17, 46, 68, 71).

[105] H. Z. Li and P. Boulanger. "A Survey of Heart Anomaly Detection Using Ambulatory Electrocardiogram (ECG)". In: *Sensors (Basel, Switzerland)* 20 (5 2020-03). ISSN: 14248220. DOI: 10.3390/S20051461. URL: /pmc/articles/PMC7085598//pmc/articles/PMC7085598/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC7085598/ (cit. on p. 51).

[106] J. Li, A. Loerbroks, and P. Angerer. "Physical activity and risk of cardiovascular disease: what does the new epidemiological evidence show?" In: *Current opinion in cardiology* 28 (5 2013-09), pp. 575–583. ISSN: 1531-7080. DOI: 10.1097/HCO.0B013E328364289C. URL: https://pubmed.ncbi.nlm.nih.gov/23928923/ (cit. on p. 18).

[107] R. T. Li et al. "Wearable Performance Devices in Sports Medicine". In: *Sports health* 8 (1 2015-11), pp. 74–78. ISSN: 19410921. DOI: 10.1177/1941738115616917. URL: https://journals.sagepub.com/doi/full/10.1177/1941738115616917 (cit. on p. 1).

[108] J. Lim et al. "ECG classification via integration of adaptive beat segmentation and relative heart rate with deep learning networks". In: *Computers in Biology and Medicine* 181 (2024-10), p. 109062. ISSN: 0010-4825. DOI: 10.1016/J.COMPBIOMED.2024.109062 (cit. on p. 40).

[109] C. Liu et al. "BioSignal Copilot: Leveraging the power of LLMs in drafting reports for biomedical signals". In: *medRxiv* (2023-07), p. 2023.06.28.23291916. DOI: 10.1101/2023.06.28.23291916. URL: https://www.medrxiv.org/content/10.1101/2023.06.28.23291916v1https://www.medrxiv.org/content/10.1101/2023.06.28.23291916v1.abstract (cit. on p. 43).

[110] G. LIU et al. "ECG quality assessment based on hand-crafted statistics and deep-learned S-transform spectrogram features". In: *Computer methods and programs in biomedicine* 208 (2021-09). ISSN: 1872-7565. DOI: 10.1016/J.CMPB.2021.106269. URL: https://pubmed.ncbi.nlm.nih.gov/34298474/ (cit. on p. 85).

[111] M. Liu and Y. Kim. "Classification of Heart Diseases Based on ECG Signals Using Long Short-Term Memory". In: Institute of Electrical and Electronics Engineers Inc., 2018-10, pp. 2707–2710. DOI: 10.1109/EMBC.2018.8512761 (cit. on p. 46).

[112] X. Liu et al. "Deep learning in ECG diagnosis: A review". In: *Knowledge-Based Systems* 227 (2021), p. 107187. DOI: https://doi.org/10.1016/j.knosys.2021.107187 (cit. on pp. 46–48).

[113] S. K. Loo and R. A. Barkley. "Clinical Utility of EEG in Attention Deficit Hyperactivity Disorder". In: *Applied Neuropsychology* 12 (2 2005), pp. 64–76. ISSN: 09084282. DOI: 10.1207/S15324826AN1202_2. URL: https://www.tandfonline.com/doi/abs/10.1207/s15324826an1202_2 (cit. on p. 1).

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

[115] C. D. Luca. "Electromyography". In: *Encyclopedia of Medical Devices and Instrumentation* (2006-04). DOI: 10.1002/0471732877.EMD097. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/0471732877.emd097https://onlinelibrary.wiley.com/doi/abs/10.1002/0471732877.emd097https://onlinelibrary.wiley.com/doi/10.1002/0471732877.emd097 (cit. on p. 8).

[116] L.-K. Lunde et al. "Heavy Physical Work: Cardiovascular Load in Male Construction Workers". In: *International Journal of Environmental Research and Public Health 2016, Vol. 13, Page 356* 13 (4 2016-03), p. 356. DOI: 10.3390/IJERPH13040356. URL: https://www.mdpi.com/1660-4601/13/4/356 (cit. on p. 17).

[117] R. Maiti. "Workload assessment in building construction related activities in India". In: *Applied Ergonomics* 39 (6 2008-11), pp. 754–765. ISSN: 0003-6870. DOI: 10.1016/J.APERGO.2007.11.010 (cit. on p. 17).

[118] D. Makowski et al. "NeuroKit2: A Python toolbox for neurophysiological signal processing". In: *Behavior Research Methods* 53.4 (2021), pp. 1689–1696. DOI: 10.3758/s13428-020-01516-y. URL: https://doi.org/10.3758\%2Fs13428-020-01516-y (cit. on pp. 77, 78, 80).

[119] Z. S. Maman et al. "A data-driven approach to modeling physical fatigue in the workplace using wearable sensors". In: *Applied Ergonomics* 65 (2017-11), pp. 515–529. ISSN: 0003-6870. DOI: 10.1016/J.APERGO.2017.02.001 (cit. on pp. 1, 14).

[120] N. R. A. Martins et al. "Fatigue Monitoring Through Wearables: A State-of-the-Art Review". In: *Frontiers in Physiology* 12 (2021-12), p. 790292. ISSN: 1664042X. DOI: 10.3389/FPHYS.2021.790292/XML/NLM. URL: www.frontiersin.org (cit. on p. 14).

[121] B. Marwan, F. Samann, and T. Schaanze. "Denoising of ECG with single and multiple hidden layer autoencoders". In: *Current Directions in Biomedical Engineering*. Vol. 8. 2. De Gruyter. 2022, pp. 652–655. DOI: 10.1515/cdbme-2022-1166 (cit. on pp. 69–71, 83).

[122] J. Matthews, J. Kim, and W. H. Yeo. "Advances in Biosignal Sensing and Signal Processing Methods with Wearable Devices". In: *Analysis and Sensing* 3 (2 2023-03), e202200062. ISSN: 2629-2742. DOI: 10.1002/ANSE.202200062. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/anse.202200062https://onlinelibrary.wiley.com/doi/abs/10.1002/anse.202200062https://chemistry-europe.onlinelibrary.wiley.com/doi/full/10.1002/anse.202200062 (cit. on pp. 12, 40).

[123] M. C. Maurer et al. "Explainable Artificial Intelligence on Biosignals for Clinical Decision Support". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2024-08), pp. 6597–6604. ISSN: 2154817X. DOI: 10.1145/3637528.3671459. URL: https://dl.acm.org/doi/10.1145/3637528.3671459 (cit. on p. 33).

[124] V. Mehlin, S. Schacht, and C. Lanquillon. "Towards energy-efficient Deep Learning: An overview of energy-efficient approaches along the Deep Learning Lifecycle". In: (2023-02). URL: https://arxiv.org/abs/2303.01980v1 (cit. on p. 35).

[125] S. Mehta et al. "ESPNetv2: A Light-weight, Power Efficient, and General Purpose Convolutional Neural Network". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2019-June (2018-11), pp. 9182–9192. ISSN: 10636919. DOI: 10.1109/CVPR.2019.00941. URL: https://arxiv.org/abs/1811.11431v3 (cit. on p. 34).

[126] S. Mekruksavanich and A. Jitpattanakul. "Hybrid convolution neural network with channel attention mechanism for sensor-based human activity recognition". In: *Scientific Reports 2023 13:1* 13 (1 2023-07), pp. 1–14. ISSN: 2045-2322. DOI: 10.1038/s41598-023-39080-y. URL: https://www.nature.com/articles/s41598-023-39080-y (cit. on p. 42).

[127] G. Menghani. "Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster, and Better". In: *ACM Computing Surveys* 55 (12 2023-12). ISSN: 15577341. DOI: 10.1145/3578938/SUPPL_FILE/3578938-SUPP.PDF. URL: https://dl.acm.org/doi/10.1145/3578938 (cit. on pp. 34, 36, 94).

[128] C. V. Mieghem, M. Sabbe, and D. Knockaert. "The Clinical Value of the ECG in Noncardiac Conditions". In: *Chest* 125 (4 2004-04), pp. 1561–1576. ISSN: 0012-3692. DOI: 10.1378/CHEST.125.4.1561 (cit. on p. 1).

[129] I. D. Mienye and T. G. Swart. "A Comprehensive Review of Deep Learning: Architectures, Recent Advances, and Applications". In: *Information 2024, Vol. 15, Page 755* 15 (12 2024-11), p. 755. ISSN: 2078-2489. DOI: 10.3390/INFO15120755. URL: https://www.mdpi.com/2078-2489/15/12/755/htmhttps://www.mdpi.com/2078-2489/15/12/755 (cit. on pp. 21, 23, 29–31, 33).

[130] K. R. Mills. "The basics of electromyography". In: *Journal of Neurology, Neurosurgery and Psychiatry* 76 (suppl 2 2005-06), pp. ii32–ii35. ISSN: 0022-3050. DOI: 10.1136/JNNP.2005.069211. URL: https://jnnp.bmj.com/content/76/suppl_2/ii32 https://jnnp.bmj.com/content/76/suppl_2/ii32.abstract (cit. on p. 8).

[131] S. Min et al. "Wearable blood pressure sensors for cardiovascular monitoring and machine learning algorithms for blood pressure estimation". In: *Nature Reviews Cardiology 2025* (2025-02), pp. 1–20. ISSN: 1759-5010. DOI: 10.1038/s41569-025-01127-0. URL: https://www.nature.com/articles/s41569-025-01127-0 (cit. on p. 12).

[132] J. Minguillon et al. "Portable System for Real-Time Detection of Stress Level". In: *Sensors 2018, Vol. 18, Page 2504* 18 (8 2018-08), p. 2504. ISSN: 1424-8220. DOI: 10.3390/S18082504. URL: https://www.mdpi.com/1424-8220/18/8/2504/htmhttps://www.mdpi.com/1424-8220/18/8/2504 (cit. on p. 13).

[133] D. Moher et al. "Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement". In: *Journal of clinical epidemiology* 62 (10 2009-10), pp. 1006–1012. ISSN: 1878-5921. DOI: 10.1016/J.JCLINEPI.2009.06.005. URL: https://pubmed.ncbi.nlm.nih.gov/19631508/ (cit. on p. 15).

[134] A. Mondal, M. S. Manikandan, and R. B. Pachori. "Convolutional Neural Network Based ECG Quality Assessment Using Derivative Signal". In: *2022 4th International Conference on Cognitive Computing and Information Processing, CCIP 2022* (2022). DOI: 10.1109/CCIP57447.2022.10058688 (cit. on p. 85).

[135] G. B. Moody and R. G. Mark. "The impact of the MIT-BIH arrhythmia database". In: *IEEE engineering in medicine and biology magazine* 20.3 (2001), pp. 45–50. DOI: 10.1109/51.932724 (cit. on p. 70).

[136] G. B. Moody, W Muldrow, and R. G. Mark. "A noise stress test for arrhythmia detectors". In: *Computers in cardiology* 11.3 (1984), pp. 381–384 (cit. on pp. 68, 71).

[137] M. Moshawrab et al. "Smart Wearables for the Detection of Occupational Physical Fatigue: A Literature Review". In: *Sensors 2022, Vol. 22, Page 7472* 22 (19 2022-10), p. 7472. ISSN: 1424-8220. DOI: 10.3390/S22197472. URL: https://www.mdpi.com/1424-8220/22/19/7472/htmhttps://www.mdpi.com/1424-8220/22/19/7472 (cit. on p. 14).

[138] N. T. Newaz and E. Hanada. "The Methods of Fall Detection: A Literature Review". In: *Sensors 2023, Vol. 23, Page 5212* 23 (11 2023-05), p. 5212. ISSN: 1424-8220. DOI: 10.3390/S23115212. URL: https://www.mdpi.com/1424-8220/23/11/5212/htmhttps://www.mdpi.com/1424-8220/23/11/5212 (cit. on p. 13).

[139] B. Neyshabur et al. "What is being transferred in transfer learning?" In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 512–523 (cit. on p. 38).

[140] B. Nowrouzi-Kia et al. "Heart Disease and Occupational Risk Factors in the Canadian Population: An Exploratory Study Using the Canadian Community Health Survey". In: *Safety and health at work* 9 (2 2018-06), pp. 144–148. ISSN: 2093-7911. DOI: 10.1016/J.SHAW.2017.07.008. URL: https://pubmed.ncbi.nlm.nih.gov/29928527/ (cit. on p. 15).

[141] E. Oliosi et al. "Correlation Between Pain Intensity and Trunk Sway in Seated Posture Among Office Workers with Chronic Spinal Pain: A Pilot Field-Based Study". In: *Sensors 2025, Vol. 25, Page 1583* 25 (5 2025-03), p. 1583. ISSN: 1424-8220. DOI: 10.3390/S25051583. URL: https://www.mdpi.com/1424-8220/25/5/1583/htmhttps://www.mdpi.com/1424-8220/25/5/1583 (cit. on p. 12).

[142] I. L. Organization. *Statistics on working time*. Accessed: 2025-01-27. 2024. URL: https://ilostat.ilo.org/topics/working-time/ (cit. on p. 1).

[143] A. Pal et al. "PHYSIOLOGICAL STRAIN AMONG WOMEN POTATO CULTIVATORS IN WEST BENGAL, INDIA". In: *Journal of human ergology* 44 (2015), pp. 61–74 (cit. on p. 17).

[144] T. Y. Pan et al. "A Hierarchical Hand Gesture Recognition Framework for Sports Referee Training-Based EMG and Accelerometer Sensors". In: *IEEE Transactions on Cybernetics* 52 (5 2022-05), pp. 3172–3183. ISSN: 21682275. DOI: 10.1109/TCYB.2020.3007173 (cit. on p. 1).

[145] K. Papoutsakis et al. "Detection of Physical Strain and Fatigue in Industrial Environments Using Visual and Non-Visual Low-Cost Sensors". In: *Technologies 2022, Vol. 10, Page 42* 10 (2 2022-03), p. 42. ISSN: 2227-7080. DOI: 10.3390/TECHNOLOGIES10020042. URL: https://www.mdpi.com/2227-7080/10/2/42/htmhttps://www.mdpi.com/2227-7080/10/2/42 (cit. on p. 17).

[146] A. Paszke et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019) (cit. on p. 73).

[147] A. Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems* 32 (2019-12). ISSN: 10495258. URL: https://arxiv.org/abs/1912.01703v1 (cit. on p. 96).

[148] J. Pfeiffer et al. "Modular Deep Learning". In: (2023-02). URL: https://arxiv.org/abs/2302.11529v2 (cit. on pp. 91, 95).

[149] G. Prieto-Avalos et al. "A Review of Commercial and Non-Commercial Wearables Devices for Monitoring Motor Impairments Caused by Neurodegenerative Diseases". In: *Biosensors 2023, Vol. 13, Page 72* 13 (1 2022-12), p. 72. ISSN: 2079-6374. DOI: 10.3390/BIOS13010072. URL: https://www.mdpi.com/2079-6374/13/1/72/htmhttps://www.mdpi.com/2079-6374/13/1/72 (cit. on p. 2).

[150] F. D. Pup and M. Atzori. "Applications of Self-Supervised Learning to Biomedical Signals: A Survey". In: *IEEE Access* 11 (2023), pp. 144180–144203. ISSN: 21693536. DOI: 10.1109/ACCESS.2023.3344531 (cit. on p. 41).

[151] N. Qian. "On the momentum term in gradient descent learning algorithms". In: *Neural Networks* 12 (1 1999), pp. 145–151. ISSN: 08936080. DOI: 10.1016/S0893-6080(98)00116-6. URL: https://pubmed.ncbi.nlm.nih.gov/12662723/ (cit. on p. 23).

[152] M. Rahman et al. "ResEMGNet: A Lightweight Residual Deep Learning Architecture for Neuromuscular Disorder Detection from Raw EMG Signals". In: (2023-09). URL: https://arxiv.org/abs/2309.10756v4 (cit. on p. 41).

[153] K. N. Rajesh and R. Dhuli. "Classification of ECG heartbeats using nonlinear decomposition methods and support vector machine". In: *Computers in biology and medicine* 87 (2017-08), pp. 271–284. ISSN: 1879-0534. DOI: 10.1016/J.COMPBIOMED.2017.06.006. URL: https://pubmed.ncbi.nlm.nih.gov/28624712/ (cit. on p. 51).

[154] V. Randazzo, J. Ferretti, and E. Pasero. "A Wearable Smart Device to Monitor Multiple Vital Parameters—VITAL ECG". In: *Electronics 2020, Vol. 9, Page 300* 9 (2 2020-02), p. 300. ISSN: 2079-9292. DOI: 10.3390/ELECTRONICS9020300. URL: https://www.mdpi.com/2079-9292/9/2/300/htmhttps://www.mdpi.com/2079-9292/9/2/300 (cit. on p. 1).

[155] M. N. Rastgoo et al. "Automatic driver stress level classification using multimodal deep learning". In: *Expert Systems with Applications* 138 (2019), p. 112793 (cit. on p. 50).

[156] B Rio, T. Lopetegi, and I Romero. "Assessment of different methods to estimate electrocardiogram signal quality". In: *Computing in Cardiology* 38 (2011-01), pp. 609–612 (cit. on p. 85).

[157] R. Rodrigues and P. Couto. "A neural network approach to ECG denoising". In: *arXiv* (2012). DOI: 10.48550/arXiv.1212.5217 (cit. on pp. 69, 77, 83, 84).

[158] S. Ruder. "An overview of gradient descent optimization algorithms". In: (2016-09). URL: https://arxiv.org/abs/1609.04747v2 (cit. on p. 23).

[159] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, 1986, 318–362. ISBN: 026268053X (cit. on pp. 24, 25).

[160] S. Saadatnejad, M. Oveisi, and M. Hashemi. "LSTM-Based ECG Classification for Continuous Monitoring on Personal Wearable Devices". In: *IEEE Journal of Biomedical and Health Informatics* 24 (2 2020-02), pp. 515–523. DOI: 10.1109/JBHI.2019.2911367 (cit. on pp. 42, 46).

[161] S. Sadasivuni et al. "Recurrent neural network circuit for automated detection of atrial fibrillation from raw ECG". In: *Proceedings - IEEE International Symposium on Circuits and Systems* 2021-May (2021). DOI: 10.1109/ISCAS51556.2021.9401666 (cit. on p. 46).

[162] E. Sajno et al. "Machine learning in biosignals processing for mental health: A narrative review". In: *Frontiers in Psychology* 13 (2023-01), p. 1066317. ISSN: 16641078. DOI: 10.3389/FPSYG.2022.1066317/PDF (cit. on p. 42).

[163] M. Sandler et al. "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520 (cit. on p. 54).

[164] A. Sathyanarayana, J. Srivastava, and L. Fernandez-Luque. "The Science of Sweet Dreams: Predicting Sleep Efficiency from Wearable Device Data". In: *Computer* 50 (3 2017-03), pp. 30–38. ISSN: 00189162. DOI: 10.1109/MC.2017.91 (cit. on p. 1).

[165] S. Schettino et al. "Forest harvesting in rural properties: Risks and worsening to the worker's health under the ergonomics approach". In: *International Journal of Industrial Ergonomics* 82 (2021-03), p. 103087. ISSN: 0169-8141. DOI: 10.1016/J.ERGON.2021.103087 (cit. on pp. 17, 18).

[166] R. M. Schmidt. "Recurrent neural networks (rnns): A gentle introduction and overview". In: *arXiv preprint arXiv:1912.05911* (2019) (cit. on p. 25).

[167] T. Schultz et al. "Biosignal-Based Spoken Communication: A Survey". In: *IEEE/ACM Transactions on Audio Speech and Language Processing* 25 (12 2017-12), pp. 2257–2271. ISSN: 23299290. DOI: 10.1109/TASLP.2017.2752365 (cit. on p. 1).

[168] R. Schwartz et al. "Green AI". In: *Communications of the ACM* 63 (12 2019-07), pp. 54–63. ISSN: 15577317. DOI: 10.1145/3381831. URL: https://arxiv.org/abs/1907.10597v3 (cit. on p. 34).

[169] N. Seeuws, M. D. Vos, and A. Bertrand. "A Human-in-the-Loop Method for Annotation of Events in Biomedical Signals". In: *IEEE Journal of Biomedical and Health Informatics* (2024). ISSN: 21682208. DOI: 10.1109/JBHI.2024.3460533 (cit. on p. 41).

[170] N. Seeuws, M. D. Vos, and A. Bertrand. "Avoiding Post-Processing with Event-Based Detection in Biomedical Signals". In: *IEEE Transactions on Biomedical Engineering* 71 (8 2024), pp. 2442–2453. ISSN: 15582531. DOI: 10.1109/TBME.2024.3375759 (cit. on p. 40).

[171] S. Senay. "Time-frequency BSS of biosignals". In: *Healthcare Technology Letters* 5 (6 2018-12), pp. 242–246. ISSN: 2053-3713. DOI: 10.1049/HTL.2018.5029. URL: https://onlinelibrary.wiley.com/doi/full/10.1049/htl.2018.5029 (cit. on p. 10).

[172] L. M. Sepulveda-Cano, C. D. Acosta-Medina, and G. Castellanos-Dominguez. "Relevance analysis of stochastic biosignals for identification of pathologies". In: *EURASIP Journal on Advances in Signal Processing* 2011 (2011), pp. 1–10 (cit. on p. 10).

[173] D. R. Seshadri et al. "Wearable sensors for monitoring the internal and external workload of the athlete". In: *npj Digital Medicine 2019 2:1* 2 (1 2019-07), pp. 1–18. ISSN: 2398-6352. DOI: 10.1038/s41746-019-0149-2. URL: https://www.nature.com/articles/s41746-019-0149-2 (cit. on p. 13).

[174] F. Shaffer and J. P. Ginsberg. "An Overview of Heart Rate Variability Metrics and Norms". In: *Frontiers in public health* 5 (2017-09). ISSN: 2296-2565. DOI: 10.3389/FPUBH.2017.00258. URL: https://pubmed.ncbi.nlm.nih.gov/29034226/ (cit. on p. 8).

[175] F. Shaffer, R. McCraty, and C. L. Zerr. "A healthy heart is not a metronome: an integrative review of the heart's anatomy and heart rate variability". In: *Frontiers in Psychology* 5 (2014-09), p. 108292. ISSN: 1664-1078. DOI: 10.3389/FPSYG.2014.01040/PDF. URL: www.frontiersin.org (cit. on p. 10).

[176] S. Shanmuganathan. "Artificial Neural Network Modelling: An Introduction". In: *Studies in Computational Intelligence* 628 (2016-02), pp. 1–14. ISSN: 1860-9503. DOI: 10.1007/978-3-319-28495-8_1. URL: https://link.springer.com/chapter/10.1007/978-3-319-28495-8_1 (cit. on p. 21).

[177] J. J. Shih, D. J. Krusienski, and J. R. Wolpaw. "Brain-Computer Interfaces in Medicine". In: *Mayo Clinic Proceedings* 87 (3 2012), p. 268. ISSN: 00256196. DOI: 10.1016/J.MAYOCP.2011.12.008. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC3497935/ (cit. on p. 13).

[178] A. I. Siam et al. "Biosignal classification for human identification based on convolutional neural networks". In: *International Journal of Communication Systems* 34 (7 2021-05), e4685. ISSN: 1099-1131. DOI: 10.1002/DAC.4685. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/dac.4685https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.4685https://onlinelibrary.wiley.com/doi/10.1002/dac.4685 (cit. on p. 13).

[179] L. Silva et al. "Respiratory Inductance Plethysmography to Assess Fatigability during Repetitive Work". In: *Sensors 2022, Vol. 22, Page 4247* 22 (11 2022-06), p. 4247. ISSN: 1424-8220. DOI: 10.3390/S22114247. URL: https://www.mdpi.com/1424-8220/22/11/4247/htmhttps://www.mdpi.com/1424-8220/22/11/4247 (cit. on p. 9).

[180] H. Sivaraks and C. A. Ratanamahatana. "Robust and accurate anomaly detection in ECG artifacts using time series motif discovery". In: *Computational and Mathematical Methods in Medicine* 2015 (2015-01). ISSN: 17486718. DOI: 10.1155/2015/453214 (cit. on p. 51).

[181] S. Śmigiel, K. Pałczyński, and D. Ledziński. "Deep Learning Techniques in the Classification of ECG Signals Using R-Peak Detection Based on the PTB-XL Dataset". In: *Sensors* 21.24 (2021), p. 8174 (cit. on pp. 48, 64).

[182] S. Śmigiel, K. Pałczyński, and D. Ledziński. "ECG Signal Classification Using Deep Learning Techniques Based on the PTB-XL Dataset". In: *Entropy* 23.9 (2021), p. 1121 (cit. on pp. 48, 64).

[183] S. Somani et al. "Deep learning and the electrocardiogram: review of the current state-of-the-art". In: *EP Europace* 23.8 (2021), pp. 1179–1191 (cit. on p. 48).

[184] S. M. Spook et al. "Implementing sensor technology applications for workplace health promotion: A needs assessment among workers with physically demanding work". In: *BMC Public Health* 19.1 (2019), pp. 1–9 (cit. on p. 67).

[185] N. Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958 (cit. on p. 31).

[186] R. J. Sternberg, ed. *Handbook of Intelligence*. Cambridge University Press, 2000-03. ISBN: 9780521593717. DOI: 10.1017/CBO9780511807947. URL: https://www.cambridge.org/core/books/handbook-of-intelligence/1CB4CF1E392B5A7332250A4DD2964DA3 (cit. on p. 20).

[187] M. Strauss et al. "Rescue Operations Lead to Increased Cardiovascular Stress in HEMS Crew Members: A Prospective Pilot Study of a German HEMS Cohort". In: *Journal of Clinical Medicine 2021, Vol. 10, Page 1602* 10 (8 2021-04), p. 1602. DOI: 10.3390/JCM10081602. URL: https://www.mdpi.com/2077-0383/10/8/1602/htmhttps://www.mdpi.com/2077-0383/10/8/1602 (cit. on p. 17).

[188] N. Strodthoff et al. "Deep learning for ECG analysis: Benchmarks and insights from PTB-XL". In: *IEEE Journal of Biomedical and Health Informatics* 25.5 (2020), pp. 1519–1528 (cit. on p. 48).

[189] I. Suarbawa, N Adiputra, and J. Pangkahila. "Work posture improvement using ergonomic approach decreases subjective disorders of perapen workers on the process of nguwad gamelan in Bali". In: *International Research Journal of Engineering, IT & Scientific Research* 2 (9 2016), pp. 9–17. URL: https://core.ac.uk/download/pdf/230596990.pdf (cit. on p. 17).

[190] D. P. Subha et al. "EEG signal analysis: a survey." In: *Journal of medical systems* 34 (2 2010-12), pp. 195–212. ISSN: 01485598. DOI: 10.1007/S10916-008-9231-Z/FIGURES/9. URL: https://link.springer.com/article/10.1007/s10916-008-9231-z (cit. on p. 9).

[191] H. Sun. "Modularity in deep learning". In: (2024-01). URL: https://theses.hal.science/tel-04418605https://theses.hal.science/tel-04418605/document (cit. on p. 91).

[192] K. Takahashi and H. Ueno. "Ballistocardial Signal-Based Personal Identification Using Deep Learning for the Non-Invasive and Non-Restrictive Monitoring of Vital Signs". In: *Sensors 2024, Vol. 24, Page 2527* 24 (8 2024-04), p. 2527. ISSN: 1424-8220. DOI: 10.3390/S24082527. URL: https://www.mdpi.com/1424-8220/24/8/2527/htmhttps://www.mdpi.com/1424-8220/24/8/2527 (cit. on p. 42).

[193] G. Tamburrini. "The AI Carbon Footprint and Responsibilities of AI Scientists". In: *Philosophies 2022, Vol. 7, Page 4* 7 (1 2022-01), p. 4. ISSN: 2409-9287. DOI: 10.3390/PHILOSOPHIES7010004. URL: https://www.mdpi.com/2409-9287/7/1/4/htmhttps://www.mdpi.com/2409-9287/7/1/4 (cit. on pp. 4, 35).

[194] C. Tan et al. "A Survey on Deep Transfer Learning". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11141 LNCS (2018-08), pp. 270–279. ISSN: 16113349. DOI: 10.1007/978-3-030-01424-7_27. URL: https://arxiv.org/abs/1808.01974v1 (cit. on p. 37).

[195] J. F. Thayer, S. S. Yamamoto, and J. F. Brosschot. "The relationship of autonomic imbalance, heart rate variability and cardiovascular disease risk factors". In: *International Journal of Cardiology* 141 (2 2010-05), pp. 122–131. ISSN: 0167-5273. DOI: 10.1016/J.IJCARD.2009.09.543 (cit. on p. 46).

[196] M. Thayumanavan and A. Ramasamy. "Deep Learning Based Stroke Disease Prediction using Multi Modal Biosignals". In: *2023 International Conference on Next Generation Electronics, NEleX 2023* (2023). DOI: 10.1109/NELEX59773.2023.10421131 (cit. on p. 40).

[197] B. H. Tracey and E. L. Miller. "Nonlocal means denoising of ECG signals". In: *IEEE transactions on biomedical engineering* 59.9 (2012), pp. 2383–2386 (cit. on p. 69).

[198] M. Tröger and R. Dengler. "The role of electromyography (EMG) in the diagnosis of ALS." In: *Amyotrophic Lateral Sclerosis and Other Motor Neuron Disorders : Official Publication of the World Federation of Neurology, Research Group on Motor Neuron Diseases* 1 Suppl 2 (SUPPL. 2 2000-06), S33–40. ISSN: 1466-0822. DOI: 10.1080/14660820052415808. URL: https://europepmc.org/article/med/11464939 (cit. on p. 1).

[199] A. Ullah et al. "Classification of arrhythmia by using deep learning with 2-D ECG spectral image representation". In: *Remote Sensing* 12.10 (2020), p. 1685 (cit. on pp. 48, 49).

[200] G. Van Rossum and F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697. URL: https://www.python.org (cit. on p. 96).

[201] A. Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017) (cit. on pp. 54, 55).

[202] A. Vaswani et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems* 2017-December (2017-06), pp. 5999–6009. ISSN: 10495258. URL: https://arxiv.org/abs/1706.03762v7 (cit. on pp. 29, 30).

[203] R. Verdecchia, J. Sallou, and L. Cruz. "A Systematic Review of Green AI". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 13 (4 2023-01). ISSN: 19424795. DOI: 10.1002/widm.1507. URL: https://arxiv.org/abs/2301.11047v3 (cit. on pp. 34, 35).

[204] S. C. Virgeniya and E. Ramaraj. "A Novel Deep Learning based Gated Recurrent Unit with Extreme Learning Machine for Electrocardiogram (ECG) Signal Recognition". In: *Biomedical Signal Processing and Control* 68 (2021), p. 102779. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2021.102779. URL: https://www.sciencedirect.com/science/article/pii/S1746809421003761 (cit. on pp. 48, 64).

[205] P. Wagner et al. "PTB-XL, a large publicly available electrocardiography dataset". In: *Scientific Data 2020 7:1* 7 (1 2020-05), pp. 1–15. ISSN: 2052-4463. DOI: 10.1038/s41597-020-0495-6 (cit. on pp. 47, 52, 71).

[206] J. Wang et al. "Adversarial de-noising of electrocardiogram". In: *Neurocomputing* 349 (2019-07), pp. 212–224. ISSN: 0925-2312. DOI: 10.1016/J.NEUCOM.2019.03.083 (cit. on pp. 46, 70, 83).

[207] J. Wang et al. "Generalizing to Unseen Domains: A Survey on Domain Generalization". In: *IEEE Transactions on Knowledge and Data Engineering* 35 (8 2023-08), pp. 8052–8072. ISSN: 15582191. DOI: 10.1109/TKDE.2022.3178128 (cit. on p. 32).

[208] X. Wang et al. "Self-Tuning for Data-Efficient Deep Learning". In: *Proceedings of Machine Learning Research* 139 (2021-02), pp. 10738–10748. ISSN: 26403498. URL: https://arxiv.org/abs/2102.12903v2 (cit. on p. 36).

[209] J Welch et al. "The Massachusetts General Hospital-Marquette Foundation hemodynamic and electrocardiographic database–comprehensive collection of critical care waveforms". In: *Clinical Monitoring* 7.1 (1991), pp. 96–97 (cit. on p. 83).

[210] R. J. Williams and D. Zipser. "Gradient-based learning algorithms for recurrent networks and their computational complexity". In: (1995) (cit. on p. 26).

[211] G. K. Wolf and J. H. Arnold. "Noninvasive assessment of lung volume: Respiratory inductance plethysmography and electrical impedance tomography". In: *Critical Care Medicine* 33 (3 SUPPL. 2005-03). ISSN: 00903493. DOI: 10.1097/01.CCM.0000155917.39056.97. URL: https://journals.lww.com/ccmjournal/fulltext/2005/03001/noninvasive_assessment_of_lung_volume__respiratory.9.aspx (cit. on p. 9).

[212] C.-J. Wu et al. "Sustainable AI: Environmental Implications, Challenges and Opportunities". In: *Explainable AI (XAI) for Sustainable Development: Trends and Applications* (2021-10), pp. 1–15. DOI: 10.1201/9781003457176-1. URL: https://arxiv.org/abs/2111.00364v2 (cit. on p. 34).

[213] B. Xia et al. "Intelligent cardiovascular disease diagnosis using deep learning enhanced neural network with ant colony optimization". In: *Scientific Reports 2024 14:1* 14 (1 2024-09), pp. 1–16. ISSN: 2045-2322. DOI: 10.1038/s41598-024-71932-z. URL: https://www.nature.com/articles/s41598-024-71932-z (cit. on p. 41).

[214] S. Yang et al. "A Multi-View Multi-Scale Neural Network for Multi-Label ECG Classification". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* (2023) (cit. on pp. 48, 64, 65).

[215] O. Yildirim et al. "A new approach for arrhythmia classification using deep coded features and LSTM networks". In: *Computer Methods and Programs in Biomedicine* 176 (2019), pp. 121–133. DOI: 10.1016/j.cmpb.2019.05.004. URL: www.elsevier.com/locate/cmpb (cit. on p. 46).

[216] S. Yuan et al. "Long-term electrocardiogram signal quality assessment pipeline based on a frequency-adaptive mean absolute deviation curve". en. In: *Appl. Intell.* 53.17 (2023-09), pp. 20418–20440 (cit. on p. 76).

[217] W. F. Zeng et al. "AlphaPeptDeep: a modular deep learning framework to predict peptide properties for proteomics". In: *Nature Communications 2022 13:1* 13 (1 2022-11), pp. 1–14. ISSN: 2041-1723. DOI: 10.1038/s41467-022-34904-3. URL: https://www.nature.com/articles/s41467-022-34904-3 (cit. on p. 92).

[218] B. Zhang, A. Davoodi, and Y. H. Hu. "Exploring Energy and Accuracy Tradeoff in Structure Simplification of Trained Deep Neural Networks". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 8 (4 2018-12), pp. 836–848. ISSN: 21563357. DOI: 10.1109/JETCAS.2018.2833383 (cit. on p. 34).

[219] J. Zhang et al. "A Signal Quality Assessment Method for Electrocardiography Acquired by Mobile Device". In: *Proceedings - 2018 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2018* (2019-01), pp. 2826–2828. DOI: 10.1109/BIBM.2018.8621160 (cit. on p. 85).

[220] X. Zhang et al. "Deep Learning-Based Signal Quality Assessment for Wearable ECGs". In: *IEEE Instrumentation and Measurement Magazine* 25 (5 2022-08), pp. 41–52. ISSN: 19410123. DOI: 10.1109/MIM.2022.9832823 (cit. on p. 86).

[221] J. Zhong et al. "Gated Recurrent Unit Network for Psychological Stress Classification Using Electrocardiograms from Wearable Devices". In: *Sensors (Basel, Switzerland)* 22 (22 2022-11). ISSN: 1424-8220. DOI: 10.3390/S22228664. URL: https://pubmed.ncbi.nlm.nih.gov/36433261/ (cit. on p. 42).

[222] M. Zhong et al. "Quality Assessment of Electrocardiogram Signals Using Contrastive Learning". In: *Proceedings - 2023 13th International Conference on Information Technology in Medicine and Education, ITME 2023* (2023), pp. 323–328. DOI: 10.1109/ITME60234.2023.00073 (cit. on p. 86).

[223] K. Zhou et al. "Domain Generalization: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (4 2023-04), pp. 4396–4415. ISSN: 19393539. DOI: 10.1109/TPAMI.2022.3195549 (cit. on pp. 32, 94).

[224] X. Zhou et al. "ECG Quality Assessment Using 1D-Convolutional Neural Network". In: *International Conference on Signal Processing Proceedings, ICSP* 2018-August (2019-02), pp. 780–784. DOI: 10.1109/ICSP.2018.8652479 (cit. on p. 85).

[225] X. Zhou et al. "Electrocardiogram Quality Assessment with a Generalized Deep Learning Model Assisted by Conditional Generative Adversarial Networks". In: *Life 2021, Vol. 11, Page 1013* 11 (10 2021-09), p. 1013. ISSN: 2075-1729. DOI: 10.3390/LIFE11101013. URL: https://www.mdpi.com/2075-1729/11/10/1013/htmhttps://www.mdpi.com/2075-1729/11/10/1013 (cit. on p. 85).

[226] H. Zhu et al. "A lightweight piecewise linear synthesis method for standard 12-lead ECG signals based on adaptive region segmentation". In: *PLOS ONE* 13 (10 2018-10), e0206170. ISSN: 1932-6203. DOI: 10.1371/JOURNAL.PONE.0206170. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0206170 (cit. on p. 52).

[227] F. Zhuang et al. "A Comprehensive Survey on Transfer Learning". In: *Proceedings of the IEEE* 109 (1 2021-01), pp. 43–76. ISSN: 15582256. DOI: 10.1109/JPROC.2020.3004555 (cit. on p. 37).

A

NeuralLib Structure

A.1 Directory Breakdown

The *NeuralLib* package is structured into several modules, each responsible for a specific aspect of biosignal processing and deep learning model management. Below is a detailed breakdown:

- **architectures/**
  - `base.py`: Defines a base class for neural architectures (*Architecture*), establishing common methods and attributes.
  - `biosignals_architectures.py`: Defines different architectures suitable for biosignal processing.
  - `train_architectures.py`: Implements training routines for models, namely for running grid search for hyperparameter tuning.
  - `upload_to_hugging.py`: Provides functionality to upload trained models on the Hugging Face platform.
  - `post_process_fn.py`: Includes post-processing functions for refining model outputs.
- **model_hub/**
  - `production_model.py`: Handles loading, saving, and managing trained models (*ProductionModel*) and using them for inference.
  - `TL.py`: Facilitates transfer learning workflows (with *TLFactory*), allowing users to extract trained components from *ProductionModels* and fine-tune them for new tasks, creating new models.
- **utils/**
  - `plots.py`: Includes visualization tools for analyzing model outputs and performance metrics.

  - `utils.py`: Defines *DatasetSequence*, a pytorch Dataset inheriting class for handling biosignal data. Contains helper functions for validating the dataset according to the task, data imports, and other general utilities, like device configuration.

- **Root-level scripts**
  - `__init__.py`: Initializes the package and ensures modularity.
  - `config.py` and `config_pip.py`: Handle configuration settings, including package dependencies and installation settings.

B

Usage Guide for NeuralLib

This appendix provides a practical guide to using `NeuralLib`, complementing the conceptual descriptions presented in Chapter 6. It is intended to help readers understand how to configure, train, evaluate, and deploy models using the library in real-world scenarios. To ensure clarity and consistency, the entire guide is structured around a running example: a GRU-based sequence-to-sequence model named `ECGPeakDetector`, designed for ECG peak detection. This model classifies each time step of an ECG signal as either a peak or a non-peak, serving as a representative case for typical biosignal classification tasks.

The guide is divided into three main sections:

- In Section B.1, the `ECGPeakDetector` model is trained from scratch using a single manually defined configuration of hyperparameters. The training workflow is broken down step by step, including model initialization, training, evaluation, and saving of outputs.
- In Section B.2, a grid search is performed to explore multiple hyperparameter configurations and identify the one yielding the lowest validation loss. The best-performing model is then tested, evaluated on a single signal, and uploaded to Hugging Face for sharing and reuse.
- Finally, Section B.3 demonstrates how to work with pre-trained models available in the model hub. A model trained for ECG denoising (introduced in Chapter 5) is used to initialize a new model through transfer learning using `TLFactory` and `TLModel`. Selected layers from the pre-trained model are reused and fine-tuned on the peak detection task.

B.1 Train a model from scratch

In the first section, the goal is to train a new model from scratch. In this example, we are training a model called `ECGPeakDetector`, for peak detection in ECG signals, as the name suggests. We will start by defining the architecture's hyperparameters and the training conditions, followed by model initialization, training, and testing.

Define architecture's hyperparameters and training conditions

```
arch_params = {
    'model_name': 'ECGPeakDetector',
    'n_features': 1,
    'hid_dim': [64, 128], # Hidden dimensions per layer
    'n_layers': 2,
    'dropout': 0.3,
    'learning_rate': 0.01,
    'bidirectional': True,
    'task': 'classification',
    'num_classes': 1, # binary classification
}

train_params = {
    'path_x': 'path_to_input_data',
    'path_y': 'path_to_output_data',
    'epochs': 100,
    'batch_size': 256,
    'patience': 20,
    'dataset_name': 'dataset1',
    'trained_for': 'peak detection',
    'all_samples': False,
    'samples': 300,
    'gpu_id': 0,
    'enable_tensorboard': True
}
```

Initialize and Train the Model

Define the model's architecture and set the hyperparameters. `list_architectures()` provides the list of available architectures. In this example, we are using a sequence-to-sequence GRU-based architecture. For performing peak detection, we will classify each timestep of the signal as either peak or no peak (binary classification). As such, `task` is set to `classification`, and has 1 class (binary classification) - the criterion (loss function) is thus automatically set to `BCEWithLogitsLoss`.

```
from NeuralLib.architectures import GRUseq2seq
from NeuralLib.architectures import list_architectures

list_architectures()

model = GRUseq2seq(**arch_params)

model.train_from_scratch(**train_params)

# checkpoints directory
checkpoints_dir = model.checkpoints_directory
print(checkpoints_dir)
```

Breakdown of `train_from_scratch()`:

1. **Model Initialization & Checkpoints**

   A directory for storing model checkpoints is created in: `<DEV_BASE_DIR>/results/<model_name>/checkpoints/<architecture_name_hparams_datetime>`

2. **Dataset & DataLoader Preparation**

   Training and validation datasets are instantiated from `DatasetSequence`, loading data from `path_x` (inputs) and `path_y` (outputs). PyTorch `DataLoader` objects are created with support for dynamic sequence lengths. Both `path_x` and `path_y` must contain the subfolders `train`, `val`, and `test` - during training, `path_x/train` and `path_x/val` are accessed; for testing, the data from `path_x/test` is used.

3. **Defining Callbacks for Training**

   - **Checkpoint Callback:** Saves the best model based on validation loss (`val_loss`).
   - **Early Stopping Callback:** Stops training early if validation loss does not improve for a number of epochs defined by `patience`.
   - **Loss Plot Callback:** Saves a loss curve to visualize training progress.

4. **Trainer Initialization & Logging**

   If TensorBoard is enabled, a `TensorBoardLogger` is created for tracking metrics and hyperparameters (`hparams.yaml`), saved in the checkpoint directory. The PyTorch Lightning `Trainer` is then instantiated with specifications for maximum epochs (`epochs`), device configuration, defined callbacks, and logging.

5. **Training Execution**

   The model is trained using:

   `trainer.fit(model, train_dataloader, val_dataloader)`

6. **Post-Training Processing & Model Saving**

   - The **best (lowest) validation loss** is retrieved from the checkpoint callback.
   - Training metadata (trainer state, optimizer, dataset used, GPU info, loss metrics, etc.) is saved via `model.save_training_information()` and written to `training_info.json` in the checkpoint directory.
   - The final model weights (corresponding to the best validation loss) are saved as `model_weights.pth` in the same directory.

Test the model on the test set

```
predictions, avg_loss = model.test_on_test_set(
    path_x=train_params['path_x'], # the test subfolder is used in this case
    path_y=train_params['path_y'],
    checkpoints_dir=checkpoints_dir,
    gpu_id=train_params['gpu_id'],
    all_samples=False, # if True, test on all available samples
    samples=5,
    save_predictions=True
)

print(f"Average Test Loss: {avg_loss:.4f}")
```

Summary

- Defined architecture and training parameters
- Initialized and trained a GRU model for ECG peak detection
- Tested the model on the test set
- Saved checkpoints and training metadata

B.2 Perform grid search and upload optimal configuration

In the present section, the goal is to perform grid search on `ECGPeakDetector` to find the hyperparameters that lead to the lowest validation loss.

Define architecture's hyperparameters options and perform grid search

`run_grid_search()` trains the model for all valid combinations of the provided hyperparameter options. In `archi_params_options`, each hyperparameter is assigned a list of possible values instead of a single value. If an incompatible configuration is encountered (e.g., specifying 2 layers and 3 hidden dimensions), it is automatically skipped without raising an error.

```
import NeuralLib.architectures as arc

architecture_name = 'GRUseq2seq'

archi_params_options = {
    'model_name': 'ECGPeakDetector', # model name remains always the same, regardless
        of the configuration. in the end, a single configuration is selected
    'n_features': [1],
    'hid_dim': [[32, 64, 64], [64, 64, 64], [64, 128, 64], [64, 128]],
    'n_layers': [3, 2],
    'dropout': [0.3, 0],
    'learning_rate': [0.001],
    'bidirectional': [True],
    'task': ['classification'],
    'num_classes': [1],
}

train_params = {
    'path_x': 'path_to_input_data',
    'path_y': 'path_to_output_data',
    'epochs': 100,
    'batch_size': 256,
    'patience': 20,
    'dataset_name': 'dataset1',
    'trained_for': 'peak detection',
    'all_samples': False,
    'samples': 300,
    'gpu_id': 0,
    'enable_tensorboard': True,
}

best_dir, best_val_loss, val_losses = arc.run_grid_search(architecture_name,
    archi_params_options, train_params)
```

Test the best model configuration on the test set

```
# Load architecture parameters from the hparams.yaml file
architecture_params = arc.get_hparams_from_checkpoints(best_dir)

# Initialize the model using the loaded parameters
model = arc.GRUseq2seq(**architecture_params)

predictions, avg_loss = model.test_on_test_set(
    path_x=train_params["path_x"],
    path_y=train_params["path_y"],
    checkpoints_dir=best_dir,
    gpu_id=train_params["gpu_id"],
    save_predictions=True,
    all_samples=False,
    samples=5,
)
```

Test the best model on a single signal

```
import os
import numpy as np

signal_dir = os.path.join('path_to_signal', 'example.npy')
signal = np.load(signal_dir)
single_prediction = model.test_on_single_signal(signal, checkpoints_dir=best_dir,
    gpu_id=train_params["gpu_id"])
```

Upload to Hugging Face the best configuration model

To upload the model to Hugging Face, it is only necessary that the three files that are uploaded are in a single folder (copy and paste them from the model's checkpoints to any folder): hparams.yaml, model_weights.pth, training_info.json. An additional file is automatically created, README.md, for the model card, which contains the basic information about the model to be presented in Hugging Face.

```
# Example
description = "GRU-based model for ECG peak detection"
arc.upload_production_model(local_dir='model_files_directory',
                    repo_name='repo_name',
                    token='insert_your_token',
                    model_name='ECGPeakDetector',
                    description=description)
```

Summary

- Defined hyperparameter search space using `archi_params_options`
- Executed grid search using `run_grid_search()` to identify the configuration with the lowest validation loss

- Loaded and tested on the test set the best-performing model
- Performed evaluation on a single input signal
- Uploaded the trained model and metadata to Hugging Face for sharing and reuse

B.3 Use the Model Hub for making predictions and perform Transfer Learning

Make predictions with a Production Model

```
import NeuralLib.model_hub as mh
from NeuralLib.architectures import post_process_peaks_binary
import numpy as np

mh.list_production_models() # Check Production Models available in the NeuralLib
    Collection

# Import the model
peak_detector = mh.ProductionModel(model_name="ECGPeakDetector")

# Make predictions
predicted_peaks = peak_detector.predict(
    X=test_signal,
    gpu_id=None,
    post_process_fn=post_process_peaks_binary, # post_process_peaks_binary: defined
        function for post-processing the output from ECGPeakDetector
    threshold=0.5, # parameter of post_process_peaks_binary
    filter_peaks=True # parameter of post_process_peaks_binary
)
```

Use TLFactory to extract pre-trained parameters for TLModel

In this part, we are importing a production model from the collection to apply Transfer Learning on a different task.

```
from NeuralLib.architectures import GRUseq2seq

factory = mh.TLFactory() # initialize the TL factory

factory.load_production_model(model_name="ECGDenoiser") # load the production model to
     the factory

prod_model = factory.models['ECGDenoiser'] # get the production model from the factory

# Define the TLModel configuration
arch_params = {
    'model_name': 'ECGPeakDetectorTLDenoiser',
    'n_features': 1,
    'hid_dim': [64, 64, 32],
    'n_layers': 3,
    'dropout': 0,
    'learning_rate': 0.001,
    'bidirectional': True,
    'task': 'classification',
```

```
    'num_classes': 1
}

# Initialize TLModel
tl_model = mh.TLModel('GRUseq2seq', **arch_params)

# Extract desired pre-trained parameters from the Production Model
layer_mapping = {
    'gru_layers.0': prod_model.model.gru_layers[0].state_dict(), # First GRU layer
        weights
    'gru_layers.1': prod_model.model.gru_layers[1].state_dict() # Second GRU layer
        weights
}

# Define freezing and unfreezing strategies
freeze_layers = ['gru_layers.0']
unfreeze_layers = ['gru_layers.1']

# Configure the TLModel
factory.configure_tl_model(
    tl_model=tl_model,
    layer_mapping=layer_mapping, # inject mapped weights from the prod model
    freeze_layers=freeze_layers,
    unfreeze_layers=unfreeze_layers
)

# Define training parameters for training TLModel
train_params = {
    'path_x': 'path_to_input_data',
    'path_y': 'path_to_output_data',
    'epochs': 100,
    'batch_size': 256,
    'patience': 20,
    'dataset_name': 'dataset1',
    'trained_for': 'fine-tuning for peak detection',
    'all_samples': False,
    'samples': 300,
    'gpu_id': 0,
    'enable_tensorboard': True
}

# Train TLModel
tl_model.train_tl(**train_params)
```

Summary

- Loaded a pre-trained model from the `NeuralLib` Model Hub
- Performed predictions on new input data with optional post-processing using `ProductionModel`
- Imported a production model into `TLFactory` for reuse in a new task via transfer learning
- Initialized a `TLModel` with new architecture and training parameters
- Mapped and transferred selected pre-trained weights to the new model
- Configured layer freezing and unfreezing strategies

- Trained the new model on a different task using `train_tl()`