



David José da Silva Aresta Belo

Master in Biomedical Engineering

Learning Biosignals with Deep Learning

Thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy in
Biomedical Engineering

Adviser: Hugo Gamboa, Auxiliar Professor, Faculty of
Sciences and Technology - NOVA University of
Lisbon

Examination Committee:

Rapporteurs: Prof. Alexandre José Malheiro Bernardino
Prof. Nuno Manuel Garcia dos Santos

Members: Prof. José Paulo Moreira dos Santos
Prof. Hugo Filipe Silveira Gamboa
Prof. André Ribeiro Lourenço

Learning Biosignals with Deep Learning

Copyright © David José da Silva Aresta Belo, NOVA School of Science and Technology, NOVA University Lisbon.

The NOVA School of Science and Technology and the NOVA University Lisbon have the right, perpetual and without geographical boundaries, to file and publish this dissertation through printed copies reproduced on paper or on digital form, or by any other means known or that may be invented, and to disseminate through scientific repositories and admit its copying and distribution for non-commercial, educational or research purposes, as long as credit is given to the author and editor.

I dedicate this to my mother, my soul and heart.

Acknowledgements

I thank the University Nova of Lisbon for having me, especially LIBPHys at Faculty of Sciences and Technology and my funding institution Fundação para a Ciência e Tecnologia and PLUX. I am also grateful to the teachers and professionals that worked with me Hugo Silva, Guilherme, and Manuel Pacheco from PLUX, Duarte, André, and Marília from Fraunhofer and Ana Fred from Instituto de Telecomunicações at Instituto Superior de Tecnico. I would also like to thank Frontier Development Lab, NASA, and all their partners for the wonderful opportunity given to me. Frank, Eleni, Brain Wong, Krittika and Brian Russel thank you so much for your optimism, effort, help, and most of all Astroskin.

I am grateful to my supervisor Hugo Gamboa which was the best superior I ever had, which taught me so much, not only as a professional and scientist but also as a human being. Nevertheless, I also thank my team João Rodrigues who is a great friend and which made working so much easier, Cátia Cepeda with her wonderful advice and energy, Daniel that taught me so much while giving me other perspectives, Ricardo that enriched within me the feeling of curiosity. I would also like to thank the Master students that helped me during my work, especially Nuno Bento and João Pestana which were outstanding students. Thank you, Paulo, Andreia, and Jorge for being good lab partners and friends.

On a more personal level, I would also like to thank my lovely wife Anna which supported me all these years by giving me great advice, emotional support, and most of all, all the love I required. Without her, it would be impossible. I would also love to thank Marcio that has helped me go through the best and worst times and held me when I needed the most. I am also very grateful for my family, all of them helped me greatly financially, emotionally, and always gave me the drive to keep going. My mother gave me so much and offered so much support, my father that provided me strength, Paula offered great advice and always pushed me further, Ana that always helped me see the best in me, Domingos that was always so warm, and Ivo that was always with me during all stages. I also would like to thank Pedro, Norbert, Sónia, and Cristina. For Beata and Filipe, thank you so much for all the help you gave me. I would like to thank my friends for accepting nothing less than excellence from me. Joana Ginjeira, João Nuno, Cristina and Raquel Lopes, Patricia, Catarina, Sara, and Lúcia are all my special friends and you provided me great support. Thank you all!

Abstract

The healthcare system, which is ubiquitously recognized as one of the most influential system in society, is facing new challenges since the start of the decade. The myriad of physiological data generated by individuals, namely in the healthcare system, is generating a burden on physicians, losing effectiveness on the collection of patient data. Information systems and, in particular, novel deep learning (DL) algorithms have been prompting a way to take this problem.

This thesis has the aim to have an impact in biosignal research and industry by presenting DL solutions that could empower this field. For this purpose an extensive study of how to incorporate and implement Convolutional Neural Networks (CNN), Recursive Neural Networks (RNN) and Fully Connected Networks in biosignal studies is discussed.

Different architecture configurations were explored for signal processing and decision making and were implemented in three different scenarios: (1) Biosignal learning and synthesis; (2) Electrocardiogram (ECG) biometric systems, and; (3) Electrocardiogram (ECG) anomaly detection systems. In (1) a RNN-based architecture was able to replicate autonomously three types of biosignals with a high degree of confidence. As for (2) three CNN-based architectures, and a RNN-based architecture (same used in (1)) were used for both biometric identification, reaching values above 90% for electrode-base datasets (Fantasia, ECG-ID and MIT-BIH) and 75% for off-person dataset (CYBHi), and biometric authentication, achieving Equal Error Rates (EER) of near 0% for Fantasia and MIT-BIH and below 4% for CYBHi. As for (3) the abstraction of healthy clean the ECG signal and detection of its deviation was made and tested in two different scenarios: presence of noise using autoencoder and fully-connected network (reaching 99% accuracy for binary classification and 71% for multi-class), and; arrhythmia events by including a RNN to the previous architecture (57% accuracy and 61% sensitivity).

In sum, these systems are shown to be capable of producing novel results. The incorporation of several AI systems into one could provide to be the next generation of preventive medicine, as the machines have access to different physiological and anatomical states, it could produce more informed solutions for the issues that one may face in the future increasing the performance of autonomous preventing systems that could be used in every-day life in remote places where the access to medicine is limited. These systems

will also help the study of the signal behaviour and how they are made in real life context as explainable AI could trigger this perception and link the inner states of a network with the biological traits.

Keywords: Biosignals; Deep Learning; Biometry; Synthesis; Abnormal ECG; Electrocardiogram;

Resumo

O sistema de saúde, que é ubiquamente reconhecido como um dos sistemas mais influentes da sociedade, enfrenta novos desafios desde o início da década. A miríade de dados fisiológicos gerados por indivíduos, nomeadamente no sistema de saúde, está a gerar um fardo para os médicos, perdendo a eficiência no conjunto dos dados do paciente. Os sistemas de informação e, mais especificamente, da inovação de algoritmos de aprendizagem profunda (DL) têm sido usados na procura de uma solução para este problema.

Esta tese tem o objetivo de ter um impacto na pesquisa e na indústria de biosinais, apresentando soluções de DL que poderiam melhorar esta área de investigação. Para esse fim, é discutido um extenso estudo de como incorporar e implementar redes neurais convolucionais (CNN), redes neurais recursivas (RNN) e redes totalmente conectadas para o estudo de biosinais.

Diferentes arquiteturas foram exploradas para processamento e tomada de decisão de sinais e foram implementadas em três cenários diferentes: (1) Aprendizagem e síntese de biosinais; (2) sistemas biométricos com o uso de eletrocardiograma (ECG), e; (3) Sistema de detecção de anomalias no ECG. Em (1) uma arquitetura baseada na RNN foi capaz de replicar autonomamente três tipos de sinais biológicos com um alto grau de confiança. Quanto a (2) três arquiteturas baseadas em CNN e uma arquitetura baseada em RNN (a mesma usada em (1)) foram usadas para ambas as identificações, atingindo valores acima de 90 % para conjuntos de dados à base de eletrodos (Fantasia, ECG-ID e MIT-BIH) e 75 % para o conjunto de dados fora da pessoa (CYBHi) e autenticação, atingindo taxas de erro iguais (EER) de quase 0 % para Fantasia e MIT-BIH e abaixo de 4 % para CYBHi. Quanto a (3) a abstração de sinais limpos e assintomáticos de ECG e a detecção do seu desvio foram feitas e testadas em dois cenários diferentes: na presença de ruído usando um autocodificador e uma rede totalmente conectada (atingindo 99 % de precisão na classificação binária e 71 % na multi-classe), e; eventos de arritmia incluindo um RNN na arquitetura anterior (57 % de precisão e 61 % de sensibilidade).

Em suma, esses sistemas são mais uma vez demonstrados como capazes de produzir resultados inovadores. A incorporação de vários sistemas de inteligência artificial em um unico sistema pederá desencadear a próxima geração de medicina preventiva. Os algoritmos ao terem acesso a diferentes estados fisiológicos e anatómicos, podem produzir soluções mais informadas para os problemas que se possam enfrentar no futuro, aumentando

o desempenho de sistemas autônomos de prevenção que poderiam ser usados na vida cotidiana, nomeadamente em locais remotos onde o acesso à medicina é limitado. Estes sistemas também ajudarão o estudo do comportamento do sinal e como eles são feitos no contexto da vida real, pois a IA explicável pode desencadear essa percepção e vincular os estados internos de uma rede às características biológicas.

Palavras-chave: Biosignals; Aprendizagem Profunda; Biometria; Síntese; Anomalias no ECG; Eletrocardiograma;

Contents

List of Figures	xix
List of Tables	xxiii
Acronyms	xxv
1 Introduction	1
1.1 Current challenges in healthcare	2
1.2 Artificial Intelligence as a Solution	2
1.3 Motivation	3
2 Human physiology and measurements	7
2.1 Biosignals	8
2.1.1 Definition	8
2.1.2 Biosensors	9
2.2 Electrophysiology of the cell	10
2.3 Cardiovascular System	12
2.3.1 Heart Physiology	12
2.3.2 Electrocardiogram	13
2.3.3 Phonocardiogram	14
2.3.4 Arrhythmias	14
2.3.5 Heart Rate Variability	14
2.3.6 Vascular System	15
2.3.7 Photoplethysmogram	16
2.4 Respiratory System	16
2.4.1 Mechanorespirogram	18
2.5 Skeleto-Muscular System	18
2.5.1 Force Mechanism	19
2.6 Nervous System	22
2.6.1 Neuron Physiology and Synapses	22
2.6.2 Information Processing	24
2.6.3 Central Nervous System	24
2.6.4 Brain Cortex structure	25

2.6.5	Sensory Processing	27
2.6.6	Memory Processing	27
2.6.7	Visual Sensing and Processing	28
2.7	Central Nervous System biosignals	29
2.8	Electrodermal Activity	30
3	Deep Neural Networks	33
3.1	History of Artificial Intelligence	33
3.2	Types of Machine Learning Methods	35
3.3	Single Layer Perception	35
3.4	Activation Functions	36
3.4.1	Sigmoid and Hyperbolic Tangent	36
3.4.2	Rectified Linear Unit (ReLU) and Leaky ReLU	37
3.4.3	Linear and Softmax	37
3.5	Deep Neural Networks	38
3.5.1	Fully Connected Network	38
3.5.2	Convolutional Neural Network	39
3.5.3	Recurrent Neural Networks	39
3.5.4	Long-Short Term Memory	41
3.5.5	Gated Recurrent Unit	42
3.6	Architecture Paradigms	43
3.6.1	Autoencoders	43
3.6.2	Reinforcement Learning	44
3.6.3	Generative Adversary Networks	44
3.7	Optimization	45
3.7.1	Backpropagation	47
3.7.2	Vanishing and Exploding Gradients	47
3.7.3	Network Initialization	48
3.7.4	Gradient Descent Algorithms	49
3.7.5	Batch Training Approaches	51
3.7.6	Cross-validation	52
3.8	Models Evaluation	53
3.9	The parallelism between real and artificial neural networks	54
3.9.1	Postsynaptic Synapses and ANN action potential	55
3.9.2	Brain Cortex and pathways	55
3.9.3	Memory Processing and Recall	55
3.9.4	Visual Sensing and Processing	56
4	State-of-the-art	59
4.1	Application of DNN	59
4.1.1	Application of Convolution Neural Networks (CNN)	60

4.1.2	Application of Recurrent Neural Networks (RNN)	63
4.1.3	Other Networks	64
4.2	State-of-the-art for the application scenarios	65
4.2.1	Biosignal Synthesis	65
4.2.2	Biometric classification	66
4.2.3	Abnormalty detection in Electrocardiogram (ECG)	68
5	Biosignal Processing for Deep Learning Architectures	69
5.1	Data Processing	70
5.1.1	Hanning Window Moving Average	70
5.1.2	Normalizing	70
5.1.3	Subsampling	71
5.1.4	Quantization	72
5.1.5	Segmentation	72
5.1.6	Spectrograms	73
5.1.7	Noise Segments Elimination	73
5.1.8	Symbolic Search in Time Series	73
6	Architectural Modules	75
6.1	CNN encoder and decoder	76
6.1.1	Filters and Stride	76
6.1.2	Sequential Concern	77
6.1.3	Decoder	77
6.1.4	Pros and cons	77
6.2	RNN decoder and encoder	78
6.2.1	Signal Quantization	78
6.2.2	Input transformation	79
6.2.3	Encoding vs Decoding	79
6.2.4	Pros and Cons	79
6.3	Dense Network	79
6.4	Latent space and fusion layers	80
7	Neural Architectures for Biosignal Processing and Decision	81
7.1	Supervised Learning	81
7.1.1	Multimodality approaches	81
7.1.2	Recurrent classification	83
7.1.3	Biosignal and time-window considerations	84
7.2	Unsupervised feature extraction with autoencoders	85
7.2.1	Transfer Learning	85
7.2.2	Encoding sequences	86
7.3	Semi-supervised Learning	86
7.3.1	Classification with pre-learned features	86

7.3.2	Active Learning	88
7.4	Biosignal Generation	90
7.4.1	RNN Generators	90
7.4.2	Agnostic Generators	90
7.4.3	GAN Generation	91
7.4.4	Mixture models	93
8	Application	97
8.1	Datasets	98
8.1.1	Fantasia	98
8.1.2	ECG-ID	98
8.1.3	MIT-BIH	98
8.1.4	FMH Cycling EMG	99
8.1.5	CYBHi	99
8.2	Learning and Synthesis of Biosignals	99
8.2.1	Methods	101
8.2.2	Results	103
8.3	Biometric Systems	107
8.3.1	ECG Biometrics	110
8.3.2	Objective for Biometrics based on ECG	111
8.3.3	Spectrogram Analysis Approach	111
8.3.4	Through Signal Morphology	114
8.3.5	Recurrent Neural Network Architecture	115
8.3.6	Temporal Convolutional Neural Network	116
8.3.7	Relative Score Threshold Classification	117
8.3.8	Results	118
8.3.9	Comparative Study	128
8.4	ECG Abnormality Detection	130
8.4.1	Methods	131
8.4.2	Datasets	134
8.4.3	Results	134
9	Conclusion and Future Remarks	143
9.1	Application Scenarios	143
9.1.1	Biosignal Synthesis	143
9.1.2	Biometric Systems	144
9.1.3	Abnormally detection	145
9.2	Contributions	146
9.2.1	List of publications	148
9.3	Future Remarks	149
9.3.1	Fusion between networks	149

9.3.2	Explainable Artificial Intelligence	150
9.3.3	AI in healthcare	151
Bibliography		153

List of Figures

2.1	Example of biosignals and corresponding sensor placement.	8
2.2	Cellular membrane structure and composition.	11
2.3	Diagram showing the phases of an action potential in relation to the membrane voltage over time and the respective ion flux during three different stages: resting state; depolarization, and; repolarization.	11
2.4	Diagram with the anatomy of the heart and phases of the cardiac cycle with the respective ECG and Phonocardiogram (PCG).	13
2.5	Diagram the artery depicting the several layers.	16
2.6	Normal Arterial pulse waveform.	17
2.7	Respiration mechanics demonstrating the muscles function.	18
2.8	Mechanorespirogram (RESP) in two different placements, chest and abdominal.	19
2.9	Composition of the muscle fasciculus, muscle fiber, myofibril and sarcolemma.	20
2.10	Example of an Electromyogram (EMG) acquired from a hamstring during a cycling exercise.	21
2.11	Example of an Accelerometer (ACC) acquired from a hamstring during a cycling exercise.	21
2.12	Diagram of a neuron.	22
2.13	Synaptic release of the transmitter proteins	24
2.14	Brain areas of the brain.	25
2.15	Histological appearance of the motor cortex.	26
2.16	Example of an acquisition of an Electroencephalogram (EEG).	29
2.17	Example of a acquisition of an system for Electrodermal Activity (EDA).	30
3.1	Artificial single neuron representation.	36
3.2	Activation functions used in Deep Neural Networks (DNN).	37
3.3	Diagram of a fully connected network	38
3.4	Diagram of a example of a CNN for object recognition.	40
3.5	Unfolded RNN.	40
3.6	Three different configurations for RNN.	41
3.7	Single unit representation of the Long Short-Term Memory (LSTM) (a) and a Gated Recurrent Units (GRU) (b) Neural Networks (NN)	42
3.8	Example of an autoencoder for reproducing images of numbers.	44

LIST OF FIGURES

3.9	Diagram for the Reinforcement Learning paradigm.	45
3.10	Diagram of a Generative Adversary Network.	46
3.11	Flow of calculus in each optimization step.	47
3.12	Sigmoid function successive derivatives graphical representation.	48
3.13	Diagram of the three batch training approaches.	51
3.14	Standard curve while training.	53
3.15	Example of a ROC curve and the point in which the Equal Error rate (EER) is calculated.	54
5.1	Example of an ECG transformed by a quantization process.	72
5.2	The SSTS is divided into three main modules - pre-processing, symbolic connotation and search.	74
6.1	Diagram of the deep learning configurations modules for biosignal processing and decision	76
6.2	Time window of a respiration signal with 42 samples and examples of filter configurations.	78
7.1	Diagram of a CNN classifier.	82
7.2	Diagram of two CNN classifiers for multi-channel biosignal approaches. . .	83
7.3	Diagram for classification using a simple RNN configuration.	84
7.4	Diagram for encoded sequences classification using RNN architecture. . . .	84
7.5	Diagram of an autoencoder for feature extraction.	85
7.6	Example of learning of a CNN autoencoder used for transfer learning. . . .	87
7.7	Diagram for classification using a simple RNN configuration.	87
7.8	Diagrams of classifiers using previous learned feature extraction.	88
7.9	Diagram of an reinforced learning mechanism for active learning.	89
7.10	Reproduction of three examples of synthetic ECGs.	91
7.11	Reproduction of four examples of real ECGs.	92
7.12	Diagram of a generator using a discriminator to create more realistic signals with an agnostic feature.	93
7.13	Diagram of the ECG Generator of Representative Encoding of Style and Symptoms (EGRESS) generator	94
7.14	Example of a synthetic ECG combining a wearable asymptomatic ECG and a clinical grade symptomatic ECG.	94
8.1	Signal pre-processing algorithm, based on character level language model. .	100
8.2	Four different acquired biosignals.	101
8.3	Sequential DNN mode.	102
8.4	ECG prediction for subject 3 of fantasia dataset.	103
8.5	Respiration Signal (RESP) prediction for subject 3 of fantasia dataset. . . .	105
8.6	ECG prediction for subject 3 of fantasia dataset.	105

8.7	History of the ECG DNN model with $S_D = 64$ and $H = 64$ for subject 10 of fantasia dataset.	106
8.8	RESP test results for error average.	107
8.9	RESP error standard deviation.	107
8.10	EMG error average.	108
8.11	EMG error standard deviation.	108
8.12	ECG error average.	108
8.13	ECG error standard deviation.	109
8.14	Two examples of spectrograms for each used database.	112
8.15	CNN Architecture for Fantasia database.	113
8.16	Example of a 5 layered dense block with a growth rate of 4.	113
8.17	Proposed biometric systems.	116
8.18	Proposed Temporal Convolutional Neural Network (TCNN) architecture. .	117
8.19	Distribution of the score of the RNN network for the predictor <i>ECG8</i> for each of the ECG signals of the Fantasia dataset.	119
8.20	Results for identification on the Fantasia dataset with the increase of batch sizes.	120
8.21	Confusion matrix for the classification of the identification of each individual of the filtered for Fantasia and RNN for the best accuracy results.	121
8.22	Confusion matrix for the classification of the identification of each individual for Fantasia and TCNN for the best accuracy results.	122
8.23	Evolution of the EER values over batch widows duration for the Fantasia dataset of the RNN model.	122
8.24	Evolution of the EER values over batch widows duration for the Fantasia dataset for the TCNN model.	123
8.25	Evolution of the mean EER values over batch widows duration for the Fantasia dataset with added Gaussian noise for the RNN method.	123
8.26	Evolution of the mean EER values over batch widows duration for the Fantasia dataset with added Gaussian noise for the TCNN method.	124
8.27	MIT-BIH identification accuracy	124
8.28	Confusion matrix for the best results for MIT-BIH with the RNN approach. .	125
8.29	Confusion matrix for the best results for MIT-BIH with the TCNN approach. .	126
8.30	Evolution of the EER values through time for the MIT-BIH dataset for the RNN method.	126
8.31	Evolution of the EER values through time for the MIT-BIH dataset for the TCNN method.	127
8.32	CYBHi identification accuracy.	127
8.33	CYBHi authentication EER values.	128
8.34	The encoder is composed by three blocks composed by a convolutional ("Conv") with a ReLU activation and a max pooling ("MP") layers.	132
8.35	Model architecture for the noise detection.	133

LIST OF FIGURES

8.36	Model architecture for the detection of different types of arrhythmia.	133
8.37	Mean Squared Error (MSE) mean and standard deviation values for the first 10 subjects within the age group of 21 to 34 years old.	135
8.38	Portion of the signal ECG 9, from the Fantasia database (a) and reconstruction of the same signal (b).	136
8.39	Normalized confusion matrix of the dataset used for the training phase, where Normal Sinus (normal signal (NS)) is the positive label and Noise Affected Sinus (noise affected signal (NAS)) is the negative label.	136
8.40	t-Distributed Stochastic Neighbour Embedding (t-SNE) representation of Normal Sinus (NS) (orange) and Noise Affected Sinus (NAS) (red) feature vectors produced by each window encoding.	137
8.41	Normalized confusion matrix of the dataset used for the testing phase, with classes Normal Sinus (NS), Electrode Motion (electrode motion (EM)), Baseline Wander (baseline wander (BW)) and Muscle Artifacts (muscle artifacts (MA)).	138
8.42	Normalized confusion matrix of the dataset used for the testing phase, with classes of Normal Sinus Rhythm (normal signal rhythm (NSR)), Paced Rhythm (Paced Rhythm (P)), Ventricular Bigeminy (Ventricular Bigeminy (B)), Sinus Bradycardia (Sinus Bradycardia (SBR)), Atrial Fibrillation (Atrial Fibrillation (AFIB)), Atrial Flutter (Atrial Flutter (AFL)), and Wolff-Parkinson-White Syndrome (Pre-excitation (PREX)).	139
8.43	Normalized confusion matrix of the dataset used for the testing phase, with classes Normal Sinus Rhythm (NSR), Paced Rhythm (P), Ventricular Bigeminy (B), Sinus Bradycardia (SBR), Atrial Fibrillation with Atrial Flutter (AFIB + AFL), and Wolff-Parkinson-White Syndrome (PREX).	140

List of Tables

2.1	Examples of Cardiac arrhythmias, description, some possible causes and ECG examples.	15
8.1	Classification performance of CNN using spectrograms.	114
8.2	Accuracy comparison for Fantasia Database	115
8.3	Accuracy comparison for ECG-ID Database.	115
8.4	Comparison with Previous Work	129
8.5	Binary noise detection model: classification performance (%)	135
8.6	Multi-class noise detection model: classification performance for each class (%)	137
8.7	classification performance (in %) for the arrhythmia detection model	140
8.8	Classification performance (in %) for the arrhythmia detection model with merged AFIB and AFL and comparison with Hannun et al. (2019) [119] . .	141

Acronyms

ACC	Accelerometer
AFIB	Atrial Fibrillation
AFL	Atrial Flutter
AI	Artificial Intelligence
ANN	Artificial Neural Networks
B	Ventricular Bigeminy
BCE	Binary Cross Entropy
BPTT	Backpropagation Through Time
BVP	Blood Volume Pressure
BW	baseline wander
CE	Cross Entropy
CNN	Convolution Neural Networks
CNS	Central Nervous System
CPU	Central Processing Unit
DL	Deep Learning
DNN	Deep Neural Networks
DTW	Dynamic Time Warping
DWT	Discrete Wavelet Transform
ECG	Electrocardiogram
ECoG	Electrocorticogram
EDA	Electrodermal Activity
EEG	Electroencephalogram
EER	Equal Error rate
EGRESS	ECG Generator of Representative Encoding of Style and Symptoms
EHR	Electronic Health Records
EM	electrode motion
EMG	Electromyogram

ACRONYMS

EOG	Electrooculogram
FFT	Fast Fourier Transform
FNR	False Negative Rate
FPR	False Positive Rate
FRR	False Rejection Rate
GAN	Generative Adversarial Networks
GPU	Graphics Processing Unit
GRU	Gated Recurrent Units
HMM	Hidden Markov Models
HRV	Heart Rate Variability
ICA	Independent Component Analysis
LSTM	Long Short-Term Memory
MA	muscle artifacts
ME	Microelectrodes
ML	Machine Learning
MSE	Mean Squared Error
NAS	noise affected signal
NN	Neural Networks
NNA	Nearest Neighbor Algorithm
NS	normal signal
NSR	normal signal rhythm
P	Paced Rhythm
PCG	Phonocardiogram
PPG	Photoplethysmogram
PREX	Pre-excitation
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RESP	Respiration Signal
RL	Reinforcement Learning
RLTC	Relative Loss Threshold Classification
RNN	Recurrent Neural Networks

ROC	Receiver Operating Characteristic
RSTC	Relative Score Threshold Classifier
SBR	Sinus Bradycardia
SGD	Stochastic Gradient Descent
SNR	Signal-to-Noise Ratio
SSTS	Syntactic Search in Time-Series
STG	Stochastic Gradient Descent
SVM	Support Vector Machine
t-SNE	t-Distributed Stochastic Neighbour Embedding
TCNN	Temporal Convolutional Neural Network
TE	Thread Electrodes
TNR	True Negative Rate
TPR	True Positive Rate
TW	time window
VAE	Variational Autoencoder
XAI	Explainable AI

CHAPTER 1

Introduction

Healthcare is one of the most powerful driving forces in empowering wealth and well-being. Furthermore, healthcare is ubiquitously recognized as having an enormous impact on society and is an important concern in every nation and culture.

Data shows that Portugal, similar to other developed countries, has seen a rise in life expectancy, access to state-of-the-art medical advances, therapeutics and innovative technology [1]. Personalized medicine, new developments in sensing devices that are becoming smaller and less invasive over time, and the global delivery systems, have improved the accessibility of healthcare. Previously, an individual's biosignals were only available in the clinical context, within limited scenarios and time of acquisition. With the conventionalization of biosignals for the general audience, the number of opportunities for the application of software capable of improving people's quality of life has increased significantly.

Electronic Health Records (EHR) are a patient-centered approach that has congregated several exams, diagnostics and historical records into the same database. In addition to the medical practitioners' accessibility to patient data, data-scientists may use it to improve healthcare status and deliver disruptive solutions [2, 3]. This trend has been reflected in an increase in healthcare investments [4]. In the European Union, the numbers have increased 9.6% between 2013 and 2016 and in Portugal 5.9%, resulting in a total of 1.35 billion and 15.5 million, respectively [5]. In the rest of the world, the current status on the expansion of China's healthcare system, India's new programs of health insurance, the increase in investment on resource-dependent countries (Middle East, Latin America, former Soviet Union), and the tax reform and health policy changes in the United States, will all play a major role in how the economics of healthcare will develop [4, 6]. The worldwide annual increase rate is estimated at 5.4% between 2017 and 2022 resulting in the expense of USD \$ 10 trillion [4].

1.1 Current challenges in healthcare

The healthcare industry is facing several challenges. According to European statistics extracted from PORDATA [5], there is roughly one physician per 300 people in Portugal, one per 280 Europeans and 310 in the Organization for Economic Co-operation and Development (OECD) countries in 2015. These numbers do not take into account each specialty. Doctors usually work several shifts and work extra hours each day and often even work back-to-back shifts. In some cases, they may stay up to 48 consecutive hours in a hospital, and, consequently, increasing the burden on doctors [7].

Another limitation of healthcare systems is that despite the increase of funding in several countries, there is often an unbalanced distribution between the rural-urban divide, remote areas and under-developed countries with low access to skilled physicians [8].

Remote access to health is also important within the context of space missions, as mankind aims to further explore our Solar System. With the 50th anniversary of the moon landing the return to lunar exploration was announced as a gateway to reaching Mars. As the human body goes to space, it undergoes numerous physiological changes, such as a decrease in muscle mass, a change in cardiovascular shape, and increased risks of cancer. Further, the astronauts are also frequently isolated which means numerous options are not available to them in their remote state. These include real-time communication and access to diagnostic machines. Currently, the use of wearable devices to monitor astronaut health is in the testing stages [9, 10].

Another challenge is the amount of information collected from devices. While systems and applications are exploiting the availability of mobile phones, wearables, and non-invasive sensors, they continuously produce information. The “curse of data” in these cases is a well-known problem as, on one hand, this information enables the possibility of understanding human health and its mechanisms and making informed decisions; on the other hand, extracting meaningful patterns from the large quantities of the available data with only the human eye and perception is impractical. Wearable devices also often increase noise and include motion artifacts that can search clean portions of the signal challenging [11].

1.2 Artificial Intelligence as a Solution

The solutions that arise for the healthcare challenges are the use of tools such as Artificial Intelligence (AI), more specifically Machine Learning (ML), to help doctors, data scientists, impoverished societies, remote inhabitants, and even astronauts to ease the burden and increase healthcare and quality of life. Since the digital revolution, the world has faced numerous changes. One of the recent digital breakthroughs was made by AI, which is presented as a branch of computer science developed to mimic and extend human intelligence. Nowadays AI and ML develop algorithms that learn the rules from a vast

amount of information and solve specific problems, in sharp contrast with traditional algorithms where humans set the rules for models to solve problems [12].

These technologies are being developed, implemented and evolving rapidly in the industry because they help professionals make their jobs easier. In 2018, a survey was performed by MemSQL [13] on 1600 respondents regarding the adoption of AI in the workplace. The survey found that 65% of respondents believe that ML was adopted to enable informed business decision making, 74% believed it to be a game-changer with the potential to transform their job and industry, 54% of these indicated that they ran models in production and 77% indicated that they were going to create new models in their short-term goals [13].

To support that AI is relevant to be implemented in healthcare, studies show that early ML adopters have achieved economic benefits [14] and the market is projected to increase in value from USD \$ 1.62 billions [15] in 2018 to USD \$ 190 billion in 2025 [16], and some state that in 2035 the additional revenue will be USD \$ 14 trillion [17]. Currently, the applications of these technologies are in the areas of image recognition and tagging, localization, mapping, maintenance predictions, security, recruitment, and the health industry. The social impact is also being considered as in 2020 it is predicted that 1.8 million jobs will be substituted by 2.3 million new jobs [18].

The digital assistants are present in our daily lives and help people with online tasks, such as schedules, answering simple questions and even control devices at home. Google, Apple, Microsoft, and Amazon have the examples of Google Assistant, Siri, Cortana, Alexa, respectively, that are expected to grow in popularity due to the dynamics of their use cases. Nowadays, 46% of North Americans use digital voice assistants [19], 72% of business executives believe that they make their lives easier [20] and the value of virtual assistants was \$5.21 billions in 2018 [21] and expected to grow to \$15.79 billion by 2021 [21]. 85% of the pioneers of the AI believe that there is an urgent need for implementing algorithms with these technologies, 90% of those pioneers have already developed strategies [22].

1.3 Motivation

Following these statistics, it is believed in the industry that AI is going to define the technologies changing the ways humans live their lives, both in the electronic world and real-life daily. The recent rise of these technologies is supported by Deep Neural Networks (DNN), a powerful ML paradigm. After the breakthrough of high performance in several sectors, such as image classification and Natural Language Processing these algorithms have had a large impact on the AI community [23]. In light of these factors, there are numerous opportunities for the application of AI such as continuous monitoring and intervention not only in healthcare but also in the sport industries, biometric applications capable of more robust results, assistants capable of interpreting our physiological states and acting accordingly by giving advice and even connecting with an intelligent system capable of

coordinating a room with the arousal of the nervous system during Intensive Care in a Heath facility.

In the face of the announced trends, this thesis proposes the contextualization of these algorithms to the biosignal area to empower systems that use biosignal processing, namely the healthcare industry. As the volume of physiological data grows new challenges arise and opportunities to have a better understanding of how biosignals work, DNN, also known as Deep Learning (DL), present themselves as a valuable tool to enhance the current status of biomedical engineering.

Currently, a complete understanding of the human systems is not accessible, where the signals come from and the true nature behind them. The signal generation that is read from the human body is tainted with all the interference that exists in the environment, right from the location where it is generated, passing through the tissues, skin, sensor, digital amplifier until the point of reception where the file is stored in the machine. With so many parameters involved in the visualization of the signal, the true nature is lost behind the interference mask.

The true machine that works behind what we observe as a signal is only an abstract notion that we can perceive but not grasp entirely. This pure notion of signal and its mechanism may contain secrets that unlock the true potential of how they can be used. Also, new study paradigms may arise where this nature is explored and biological notions may be better understood.

The creation of algorithms for each biosignal could help the integration of different types and create systems that interconnect each one delivering a more informed decision and how different types are correlated to each other. The study of the human body as a whole will assist in understanding how it works. Even though this understanding may be too ambitious to be obtained in a short period, this thesis compromises in creating a step forward in unveiling these mechanisms. By creating AI architectures that can learn with the least human intervention produce its features and the demonstration of different angles will assist in moving towards the mentioned reality. The hypothesis behind its creation may be triggered through learning and then by trying to mimic the generation of these signals through synthesis. Using this learned content may be used to discover differences between the source that generated the biosignal. And at last, the notion of what is a healthy signal, may discover all different kinds of changes to it and deliver solutions to unknown pathologies.

This thesis is comprised of nine chapters. In the second chapter, an overview of human physiology and the acquired biosignals will be made, while in the third fundamental concepts about AI and, more specifically DL, will be presented. The fourth chapter will contain the review in literature about the application of DNN in the biosignal field. The fifth, sixth and seventh chapters will depict how to preprocess biosignals specifically to be fed into DL architectures, the blocks that will build the architectures and several possible architectures to be implemented for biosignal processing and decision making, respectively. Furthermore, the eighth chapter will present the application of some of these architectures

in the field of biometry, signal synthesis, and abnormality detection. Finally the next will be a discussion, conclusion and considerations for future work and how this field could expand further.

Human physiology and measurements

Physiology is a field that studies the physical and chemical factors responsible for the origin, development, and progression of life. In human physiology the specific mechanisms and characteristics that ensure our life, by a large number of intricate control systems for breathing, motion, being warm, sensing hunger and thirst, amongst other examples. The human being is like an automaton with sensors, it feels and with the knowledge acquired it can take action upon the inputs that are given, allowing our survival in the most hostile places and demanding situations [24]. Each organ and tissue is a composition of specialized cells that are joint together by inter-cellular supporting structures. A system is composed of a network of organs that work together toward a specific body function. The human body is composed of the cardiovascular, digestive and excretory, endocrine, integumentary, immune and lymphatic, muscular, skeletal, nervous, urinary, and reproductive systems. Homeostasis is the maintenance of constant conditions and depends on the coordination between each individual element and system [24]. The search for understanding these systems gives humankind the drive to create sensors that measure them and the inspiration to develop new types of technology.

Human physiology can be measured using sensing devices that provide us a deeper understanding of their mechanisms. This chapter offers a basis of each biomedical signal behind each physiological phenomena that will give support for the contents presented throughout this thesis. The nervous system will be explained with more detail to also give the reader insights into the inspiration behind the creation of Artificial Neural Networks (ANN).

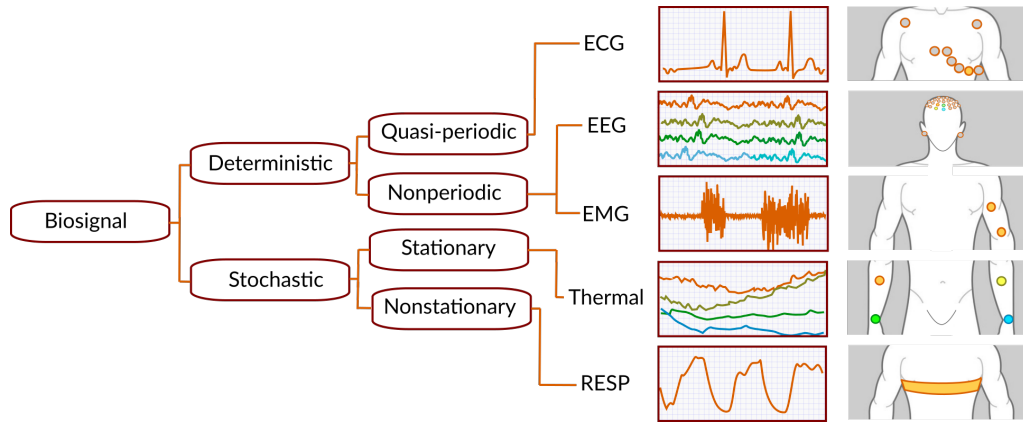


Figure 2.1: Example of biosignals and corresponding sensor placement. Adapted from [25]

2.1 Biosignals

Biosignals can be any signal which can be measured and monitored from living beings [14]. Therefore, these signals may be acquired from a visual inspection or from sensors that acquire them. A large number of different kinds of measurements reflect the numerous detections.

2.1.1 Definition

Biosignals can be grouped according to several domains. Fig. 2.1 depicts several examples of biosignals grouped according to their time-domain expression. They can be deterministic exhibiting quasi-periodic behavior, where the wave displays cycles of almost repetitive activity, and nonperiodic where the behavior does not follow this rule. The stochastic ones can be stationary, i.e. slow signals that exhibit low changes over time, and non-stationary which are dynamic are faster, displaying a large quantity of information in a short amount of time [25].

Biosignals can also be grouped according to the nature of the signal [14]:

- Electric, magnetic, or mechanic;
- Optic or Acoustic;
- Chemical or Thermal;
- Other.

The electric biosignals reflect the electrical nature of the activation of neurons or muscles while the magnetic convey information about the magnetic fields induced by these currents. The mechanic represents the mechanic changes and deformations of a system, such as relaxation and contraction of the thoracic cage during breathing. The optical reflects the absorption and scattering of the light in the body, such as the peripheral capillary oxygen

saturation (SpO_2) that is given by the variations in the blood's color, while the acoustic is the sound that is emitted by a physiological trait, such as the heartbeat result from the closing of the heart valves. The chemical biosignals are changes in the composition of body solids, liquids and gases over time, such as substances present in sweat. As for thermal, as the name implies, the changes in the absorption and emission of heat. The other class exists because not all the origins are presented here, for example, some researchers consider the human-computer interaction as biosignals due to the inference of the nervous system through the monitoring of behavioral traits captured from the keyboard or mouse movement [14].

2.1.2 Biosensors

The acquisition of these signals relies on biosensors that comprise of receptors, transducers, and amplification. Biosensors can be classified into four categories according to their physical sensing principle [26]:

- Optical
- Piezoelectric
- Thermal Sensors
- Electrochemical

One of the first optical chemical sensors was produced to measure the concentration of carbon dioxide and oxygen based on the absorption spectrum. A variety of optical sensors exist, such as spectroscopy, interferometry, and surface plasmon resonance [26].

The piezoelectric effect is observed when pressure is applied to a dielectric material. These materials convert the mechanical forces in electrical current (and the opposite as well), therefore sensors made with these materials are sensitive to changes on mass, density, and viscosity of the active area, making them capable of the detection of acoustic waves and changes of volume. The main advantage of these sensors is that they measure several domains of sensing and have a good temperature stability [26].

The thermal sensors measure the evolution of heat and are based on a thermostat or thermopile. The most common materials are metal oxide for the former and ceramic semiconductors for the last. These sensors are easy to integrate small devices due to its capacity of miniaturization [26].

The electrochemical transduction is kept in direct contact with the element and depends on a biological recognition element. The connection needs to be with three different sensors: the working electrode; the counter electrode, and; the reference electrode. They only detect reactions in the proximity of the electrode surface and the quality of acquisition depends on the properties of the electrodes [27].

The electrophysiological signals have been detected for a long time through gel-based silver-silver chloride (Ag/AgCl) electrodes. They are disposable but require the use of

gel between the electrode and the skin. The transducers transform the electrochemical changes that occur in the skin surface into electric currents for amplification and signal conditioning. These changes may cause interference causing baseline drift. Other problems that may arise are the contamination from other sources within the human body, such as the muscular activity, which may have a sufficiently high amplitude or closeness to the sensor. Major issues related to these sensors are the discomfort of using them and when there is electrode displacement making these electrodes unsuitable for daily practice [27].

Other types of these electrodes are the use of "dry" electrodes that do not require gel or conductive paste. The surface metal is non-irritant, such as stainless steel, but require careful skin preparation, such as abrasion, and is very sensitive to substances that change the skin resistivity, such as sweat or skin creams [27].

Current wearables also use capacitance sensors. These have a resistive coupling the skin to the potential capacitively through a thin insulating layer. Since the signal fidelity does not rely on good contact, these electrodes do not need preparation for the area of contact. The resistance is also stable but they suffer from movement artifacts and electrode displacements. But currently, these are the most promising for healthcare applications [27].

Currently, the fabrication of paper electrodes, meshed with gold, indium tin oxide, and carbon electrodes matrices has been researched [26, 28, 29].

2.2 Electrophysiology of the cell

A typical cell is comprised of two major parts that are the nucleus and the cytoplasm and is separated from the outside by the plasma membrane. The cell substances are called a protoplasm, which contains water and electrolytes, besides proteins, lipids, and carbohydrates. The water is present with a ratio of 70 to 85 % in most cells and many cellular chemicals are dissolved in it. The electrolytes are mostly ions such as potassium, magnesium, phosphate, sulfate, bicarbonate, sodium, chloride and calcium [24].

The cell membranes are composed of a double-layered film of phospholipid with several globular protein molecules and cholesterol. One end of each phospholipid molecule is hydrophobic, while the other end is hydrophilic. The structure depicted in Fig. 2.2 ensures that the control of the impermeability to water and other ions is made by the protein structures. Many of these are channels that permit the diffusion of selected ions or other soluble substances [24].

The cell electrical signals origin is based on the discharge of the electrical potential that exists in most of the cells of the body. The membrane is exposed to an electrochemical differential between the internal and external sides of the cell. The nerve and muscle cells are even capable of generating impulses by rapidly changing the polarity of the environment. More specifically, in the case of a nerve cell, this is caused by a mechanism due to the divergence of the potassium concentration. As the membranes are permeable to these

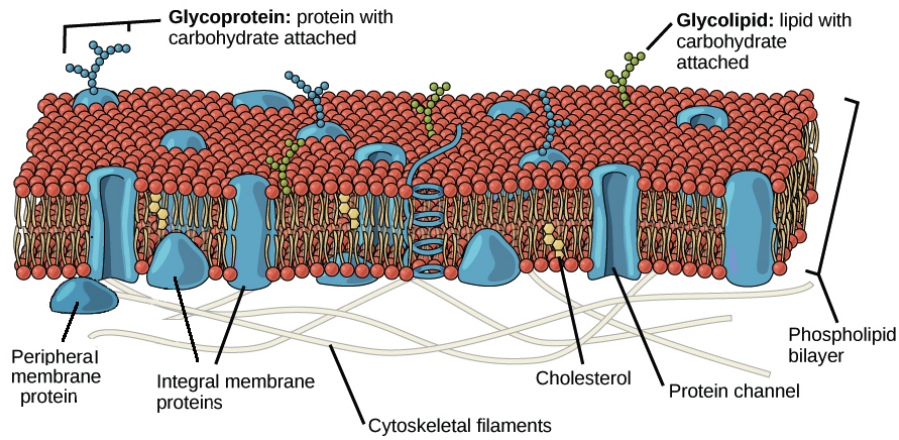


Figure 2.2: Cellular membrane structure and composition. From [30].

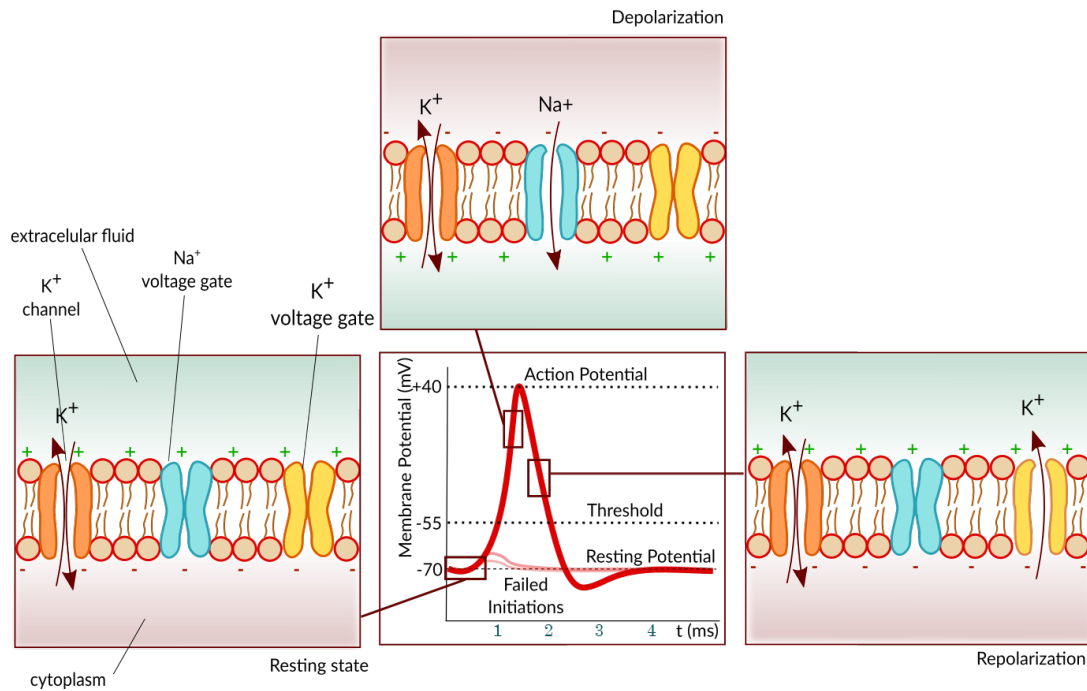


Figure 2.3: Diagram showing the phases of an action potential in relation to the membrane voltage over time and the respective ion flux during three different stages: resting state; depolarization, and; repolarization. Adapted from [24].

ions, but not permeable to other ions, the equilibrium is made by the diffusion of the potassium from the inside (which has higher concentration) to the outside (which has lower concentration) of a nerve cell (Fig. 2.3).

At the start of the cycle, the gradient is very high, causing a strong pressure of positive charge ions to pass through the membrane, causing the electric polarization to invert. The negative ions do not travel due to the permeability. This causes a potential difference between the outside and inside of the cell reaching at least 94 millivolts, in the case of a normal mammalian nerve fiber, to fire a current through the membrane. Sodium ions are also used to achieve this but oppositely. As the membrane is permeable to all other ions, for the exception of sodium, resulting in a depolarization of the membrane as the negative ions circulate from the outside to the inside within milliseconds [14, 24].

The Nernst potential is the diffusion potential level across the membrane and its magnitude is determined by the ratio of the concentrations of the specific ion that is responsible by the differential on the two sides of the membrane. The greater the value, the higher is the tendency of diffusion in one direction [14].

2.3 Cardiovascular System

The human body is full of tubes, with many different cross-sections and mechanical properties. Blood needs to circulate around our body, being pumped by the heart and spreads throughout the body using a complex system of arteries, which progressively become thinner until forming the capillary bed, where metabolic exchanges occur, and return to the heart through the venous system. The walls of the capillaries are permeable to small substances, and most are transported by diffusion due to the kinetic motion of the molecules. The heart is a muscle that is located close to the midline of the thoracic cavity and contains four chambers, two auricles, and two ventricles. The auricles are connected to the ventricles through valves, but the left side is isolated from the right side, keeping the bloodstreams from the long and short circulations separated. The right side is responsible for maintaining the circulation that provides the oxygenation of the blood (small circulation), while the left side provides oxygen to the rest of the tissues (long circulation) [31].

These physiological phenomena have an impact on how the heart behaves and how the blood pressure changes through time. Several medical diagnostics and inferences on the overall health state can be made with close inspection of the biosignals they produce.

2.3.1 Heart Physiology

In the small circulation, the blood flows from the right ventricle to the lungs, and from the lungs to the left auricle. Then, the left auricle directs the blood to the left ventricle which in turn contracts and expels the blood to the aorta and to rest of the arterial system, starting the great circulation.

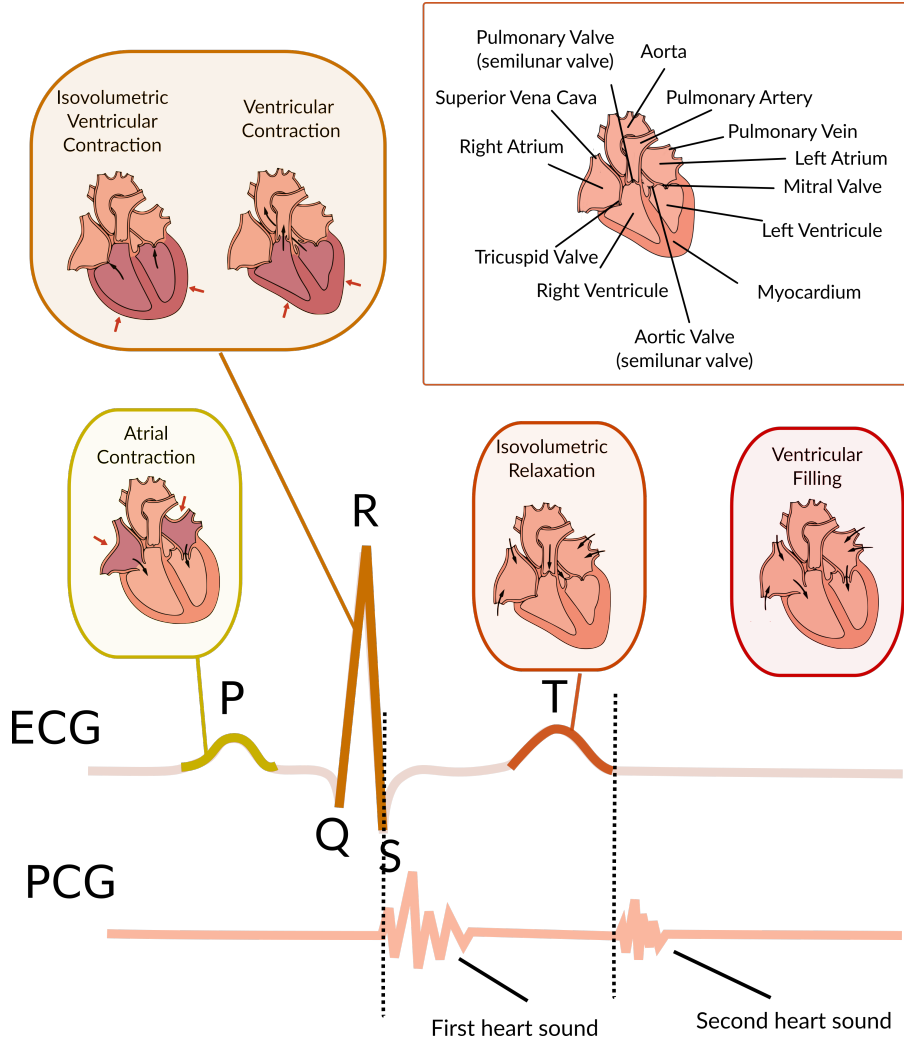


Figure 2.4: Diagram with the anatomy of the heart and phases of the cardiac cycle with the respective ECG and PCG.

The myocardium has a sequence of mechanical cyclic behaviors that are interrelated with the electrical stimulation, that can be perceived by the Electrocardiogram (ECG) signal. This cycle may be divided into the diastolic and systolic phases (Fig. 2.4). During the diastole, the ventricles are relaxed allowing the blood to come through the tricuspid (between the right auricle and ventricle) and mitral (between the left auricle and ventricle) valves, filling the cavities. These valves opened due to the increase in pressure from the contraction of the auricle. As the systole phase starts the isovolumetric contraction of the ventricles increases the pressure that after a threshold value will eject the blood through the pulmonary and aortic valves into the respective arteries [32, 33].

2.3.2 Electrocardiogram

These repeating patterns of the heart dynamics are initiated and controlled by the sinoatrial node through the propagation of an electrical current across the membranes. Each

heartbeat starts when this node creates an action potential that is conducted through the atria at 1m/s. This polarization (P wave) induces the contraction of both atria almost simultaneously. As the current is conducted through the atrioventricular node it undergoes a small time delay that enables the ejection of all the blood into the ventricles (P-Q segment). The electrical stimulus is faster than the mechanical pumping action. The electrical signal is spread through the ventricular Purkinje fibers (at approximately 1-4 m/s) at a high rate producing a ventricular contraction (QRS complex) on both sides ejecting the rest of the blood to the respective arteries [14]. The T wave represents the repolarization of the membrane when all the heart is relaxed and prepared for the next cycle. Figure 2.4 depicts pressure and blood volume changes and two biosignals that can be acquired from the heart-cycle: the ECG and the PCG.

2.3.3 Phonocardiogram

The PCG is an acoustic signal that is synchronous with the mechanical pumping action of the heart, namely the activity of the heart valves. This is affected by the pressure difference between the sides of a valve, which forces them to close or open. If the pressure is higher from the upstream, then they will open, in the opposite way, if the downstream is higher then they close [14].

The first heart sound is related to the atrioventricular valves (see Fig. 2.4), when they close, right after the R wave of the ECG. This noise is also due to the vibrations of the myocardium and deceleration of blood. In the case of the second sound is due to the closure of the valves between the ventricles and the respective arteries. These valves, which have a semilunar shape, close at the same time and the sound begins at the end of the T wave of the ECG [14].


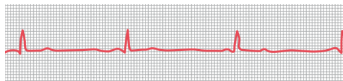



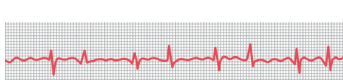
2.3.4 Arrhythmias

The heart malfunction may come from abnormal activity of the heart muscles. The deviation from the normal rhythm can give rise to arrhythmias and can be caused by changes in the pacemaker, issues regarding the pathway of the impulse or generation of spurious impulses in other parts of the heart. Table 2.1 depicts the examples of tachycardia, bradycardia, sinus arrhythmia, atrioventricular block, premature contraction, ventricular fibrillation, atrial fibrillation, and atrial flutter. Because of the physioelectrical nature of these events, all of them can be detected by analyzing the ECG.

2.3.5 Heart Rate Variability

By calculating the distance between the R peaks, the R-R signal is extracted. By analyzing the resulting graph one may understand how the heart rate behaves through time. Due to the relation with the autonomic nervous system, Heart Rate Variability (HRV) is one of the most used biosignals to diagnose not only several cardiac conditions but also nervous

Table 2.1: Examples of Cardiac arrhythmias, description, some possible causes and ECG examples. All the images are from [24]

Arrhythmia	Description	ECG example
Tachycardia	Heart beat >100 beats/min in an adult. Causes: higher body temperature; arousal of the sympathetic nerve system; toxic conditions.	
Bradycardia	Heart beat <60 beats/min in an adult. Causes: athlete's heart; vagal stimulation.	
Premature Contraction	Contraction of the myocardium before expected, named ectopic beat. Causes: ischemia; irritation caused by calcified plaques or toxics.	
Ventricular Fibrillation	Unruly ventricular impulses Causes: electrical over-stimulation; ischemia of the myocardium or pathway.	
Atrial Fibrillation	Unruly atrial impulses. Cause: Atrial enlargement, thus increased conductive pathway;	
Atrial Flutter	High rate of atria contraction, desynchronization of both atrias, erratic impulses in the A-V node. Causes: circus movement in the atria.	

system pathologies. Common HRV analysis comprises the application of a variety of analysis methods that reflect the frequency analysis of the signal. Typical components that are extracted are the high frequency (HF - 0.4–0.15 Hz), low frequency (LF - 0.15–0.04 Hz), very low frequency (VLF - 0.04–0.003 Hz), and ultralow frequency (ULF - lower than 0.003 Hz).

Each of these frequency bands reflects the activity of the parasympathetic system. HF is associated with breathing and can be determined by the connection and influences of the vagus and the sinus node. LF is related to the sympathetic vascular tone regulation center. The VLF is defined by the suprasegmental regulation of the heart rate, and the amplitude is correlated with the psycho-emotional strain and functional state of the cortex. The other rhythms are still being studied to connect with the origin and regulation factors. Even though there are correlations between the increase of VLF in cases of the central nervous system damage, anxiety, and depression disorders [34].

2.3.6 Vascular System

The arterial system may be described as a system of tubes that begins in the aorta and ramifies into a series of arteries and arterioles, with gradually small cross-section and thickness, before reaching the capillary bed. The large vessels (Fig. 2.5) are composed of

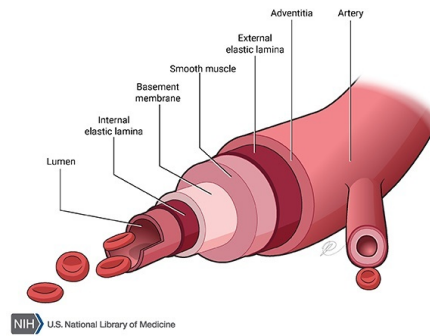


Figure 2.5: Diagram the artery depicting the several layers. From [36]

a muscular layer, the *tunica media*, and an outer connective tissue layer, named as *tunica adventitia* containing *vasa vasorum* and nerves. The capillary bed is composed only by endothelial cells and periodically by pericytes [32, 35], allowing the exchanges between the interstitial space and the blood.

Due to the elastic and muscular properties of the large arteries, they are responsible for both storing and delivering blood, during the diastole, at high pressure into the peripheral vascular beds, maintaining the flow at a constant rate during both cycles. The muscular smaller arteries modulate the pressure and velocity of the blood.

2.3.7 Photoplethysmogram

The pulse wave is detected with the Photoplethysmogram (PPG), also named as Blood Volume Pressure (BVP), which is an optical signal that reads the changes of volume of the blood in the blood vessels concerning its light absorption. The pulse wave (Fig. 2.6) is created by the contraction of the heart and the interaction of the blood with the impedance of the aorta. This wave is propagated distally through the downstream vascular system interacting with areas of varying impedance, such as the smaller arteries and arterioles, due to smaller cross-section and mechanical vascular wall properties. The narrowing of the arteries leads to an amplification of the pulse wave, leading to light reflections that occur inside these vessels of the forward propagated wave. These waves have an added effect that is propagated towards the heart and returns to the central aorta between the systole and diastole [35].

2.4 Respiratory System

The respiratory system is coordinated with the cardiovascular system. The functions of the respiratory system include regulation of mechanical ventilation, gas exchanges with blood, and, ultimately, oxygen consumption in cells to produce energy. Other functions include the adaptation of the inhaled air to the same temperature and humidity as the body core, the filtration of small particles with mucus and maintain ventilation even during food intake [14, 24].

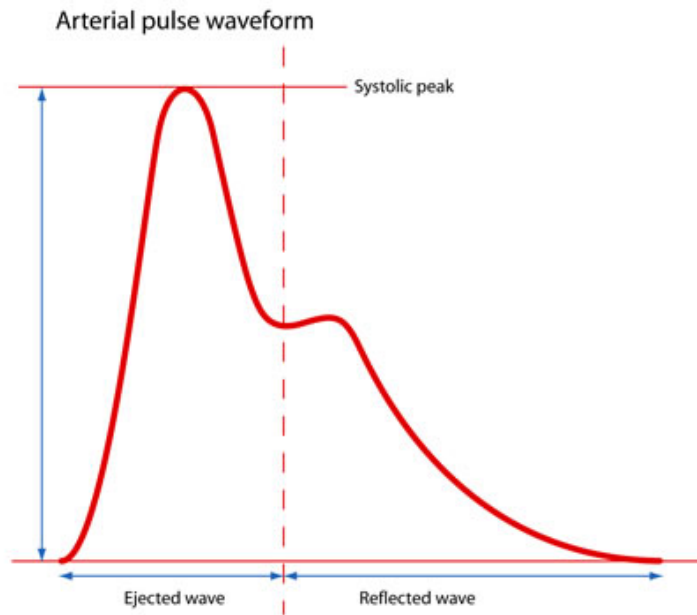


Figure 2.6: Normal Arterial pulse waveform.

The anatomy of this system starts on the upper airways, which includes the nasal and oral cavities and are merged by the uvula. The subsequent areas of breathing channel are the pharynx, trachea, bronchi, bronchioles and, finally, the alveoli. While the trachea is a tube made up of connective tissue surrounded by smooth muscles and the bronchi and bronchioles a network of progressively smaller airways, the alveoli are thin-walled sacs that are covered by a matrix of capillaries. These increase the surface area needed for efficient gas exchange, releasing oxygen to the blood and receiving blood byproducts, such as dioxide carbon, transported by diffusion. The lungs envelop the bronchi, bronchioles, and alveoli and are connected to the thoracic wall through the pleural membranes, which provide a lubricated interface. Each lung has two thin membranes that are adhered to each other supporting the sliding motion while the respiration is in effect [14]. The breathing mechanism is depicted in Fig. 2.7. The lungs have high compliance (the ability of an organ to distend in response to applied pressure) and elasticity that accommodates their forced volume increase when the muscles relax, then the passive elastic recoil reduces the volume. This mechanism is made from two sources: by the elevation and depression of the ribs, and; an upward and downward movement of the thoracic diaphragm. This skeletal muscle is connected to the inferior part of the lungs, separating the thoracic and the abdominal cavities [14]. Normal breathing is maintained mostly by this muscle, but during heavy breathing, the abdominal muscles aid the rapid expiration, by contracting against the bottom of the diaphragm [24]. During the rib cage method, the expansion is made by the projection of the sternum forward, away from the spine. The contraction is made while the ribs slide downward allowing the sternum to fall toward the vertebral column. The inspiration chest cage muscles are the external intercostals while the muscles of expiration

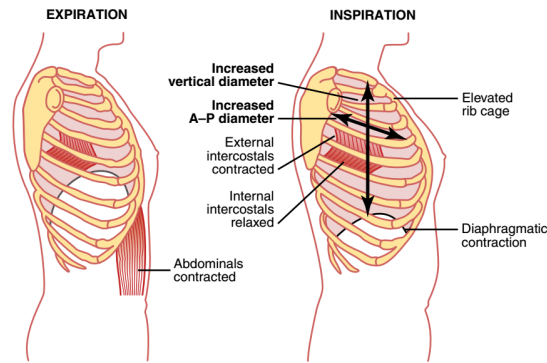


Figure 2.7: Respiration mechanics demonstrating the muscles function. From [24]

are mainly the abdominal recti and internal intercostals. These mechanisms and forces are presented in Figure 2.7 [24].

The rhythm of breathing is under both voluntary and involuntary control based on the information given by body sensors to the nervous system. These include stretch-sensitive mechanical receptors, in the lungs, chemical receptors in the brainstem, aorta, and carotid. The last is sensitive to changes to the blood pH, which indicates the levels of oxygen and carbon dioxide. The normal respiratory rate is from 10 to 20 per minute and the volume of about half a liter, increasing to 5 liters during deep breathing.

2.4.1 Mechanorespirogram

Respiration can be measured with mechanorespirogram (MRG or RESP). This captures the mechanical changes of the abdominal and the chest. In Fig. 2.8 a periodic waveform observed during normal breathing is depicted. When holding breathing that waveform turns into a stationary line. The abdominal and rib cage respiration differ in terms of amplitude, correlated with the forces and in behavior. The slightly asynchronous motion is evident when both are compared, as the abdominal is followed by the other. Sneezing, snoring, sleep apnea, are examples of conditions that may be diagnosed while analyzing the patterns of these biosignals.

Other biosignals that can be correlated with the respiration function is the thermorespirogram. The placement in the nostrils indicates the airflow direction due to the changes in its temperature. The inspiration reduces the temperature as the expiration as the opposite effect. Electroplethysmogram is also an electric biosignal that measures the thoracic impedance changes during the respiration cycle.

2.5 Skeleto-Muscular System

The muscles and skeleton provide the sustainability, stability, and motion of the human body. The contraction and relaxation of the muscles provided by the ionic pathways and electro-chemical mechanics contribute to the homeostasis by the possibility of ingesting

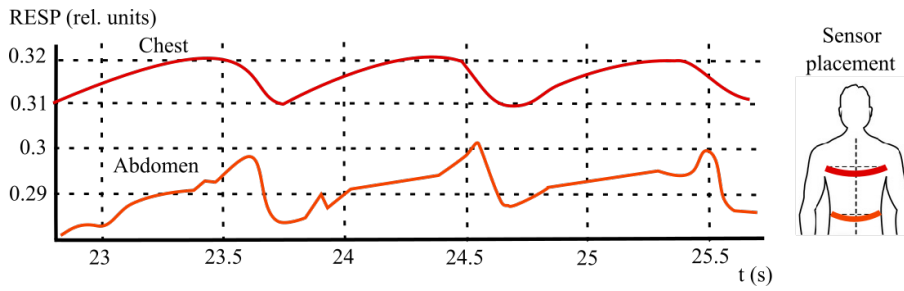


Figure 2.8: Mechanorespirogram (RESP) in two different placements, chest and abdominal. Adapted from [14]

food, water, and protection against adverse surroundings. Therefore, approximately 40% of the body is skeletal muscle and the rest divided into smooth and cardiac muscles [24].

The skeletal muscle (Fig. 2.9) is composed of fibers ranging from 10 to 80 micrometers in diameter, which can be decomposed in a hundred to thousand myofibrils. Most of these fibers are fused with the sarcolemma in the ends and have a nerve ending in the middle. The sarcolemma has an outer coat of numerous collagen fibrils and makes the connectivity between the bone and the muscle bundles of fibers, named muscle tendons [24].

The myofibrils are composed of actin and myosin filaments in a ratio of 2:1, which are composed of protein molecules. Both filaments are interdigitated with z discs, composed by different filamentous proteins, passes clockwise across the myofibril and from one to another myofibril, attaching them together. These connections give the striated appearance of the skeletal muscle. The sarcoma (myofibril between two successive z-discs) overlaps with the actin filament once the muscle is contracted, reaching only 2 micrometers. This contraction generates force [24].

2.5.1 Force Mechanism

The muscle fibers contract following a chain of actions. When the action potential is propagated through a motor nerve reaches the ending, it stimulates the release of acetylcholine, a neurotransmitter substance. This chemical open multiple gates, specialized in receiving this transmitter, localized in the muscle fiber membrane. These gates turn the membrane permeable to sodium ions that diffuse to the interior of the cells, causing the depolarization of the membrane. The action potential is transmitted through the fiber depolarizing the center of the muscle fiber causing the sarcoplasmic reticulum to release a large number of calcium ions. These ions cause the actin and myosin filaments to produce attractive forces causing them to slide alongside each other. After a few milliseconds, the calcium is pumped back to the sarcoplasmic reticulum, renovating the possibility of a subsequent contractions [24, 37].

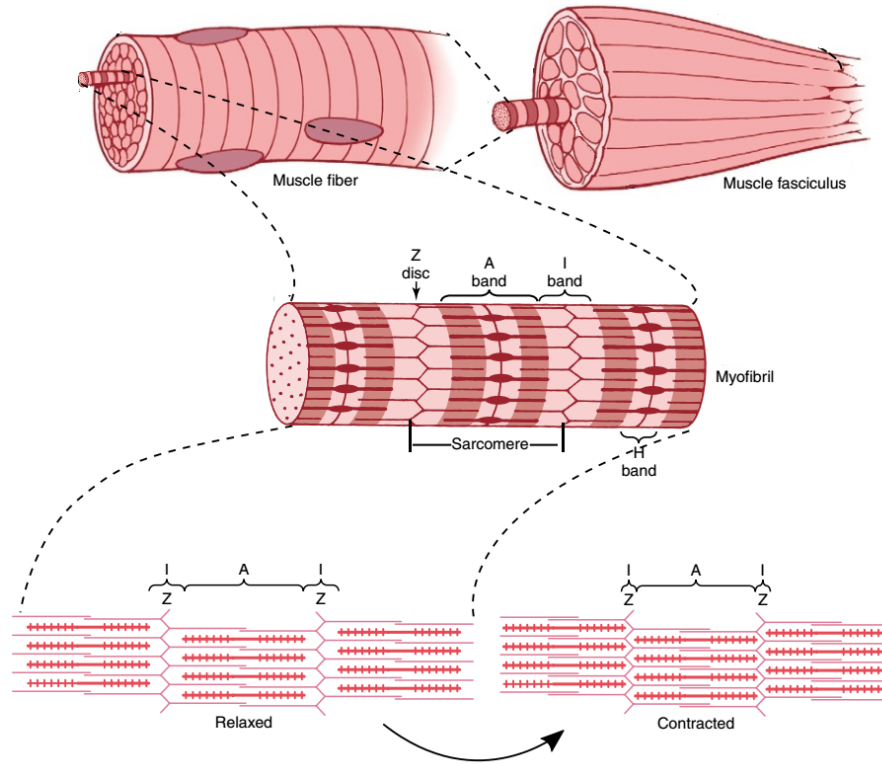


Figure 2.9: Composition of the muscle fasciculus, muscle fiber, myofibril and sarcolemma. On the bottom the mechanics of the muscle contraction: As the fiber is excited, the actin and myosin filaments to overlap. Adapted from [24]

2.5.1.1 Electromyogram (EMG)

The electrical activity of the neuromuscular activation, also named myoelectric signals, can be measured using the EMG. This random-like signal, depicted in Fig. 2.10, is based on this depolarization and repolarization mechanism. The extent of the excited zone is approximately $1\text{-}3\text{mm}^2$ and the signal is transported with a velocity of $2\text{-}6\text{m/s}$ with the muscle fiber. The cycle of each depolarization-repolarization is named an electrical dipole. The bipolar sensor configuration is paramount to understand which fibers are being affected by the measurements due to the longitudinal pathway of these signal travel along the muscle fiber. Since the motor unit depends on several muscle fibers, the electrode pair captures the magnitude of the electrical stimulus depending on their spatial distance and resolution. Therefore this signal also depends on the fiber/electrode orientation ratio [37].

The superimposition of the excitation of the motor units, named interference pattern, is observed as the asymmetric distribution of positive and negative values. Their magnitude and density are highly dependent on the firing frequency and fiber recruitment. These properties are control mechanisms for modulation and adjustment made by the nervous system. The random-like aspect of the signal is due to the constant changes in the spatial availability of the motor units [37].

Electrooculograms (EOG) is a class of EMG that are placed on the surface of the fibers

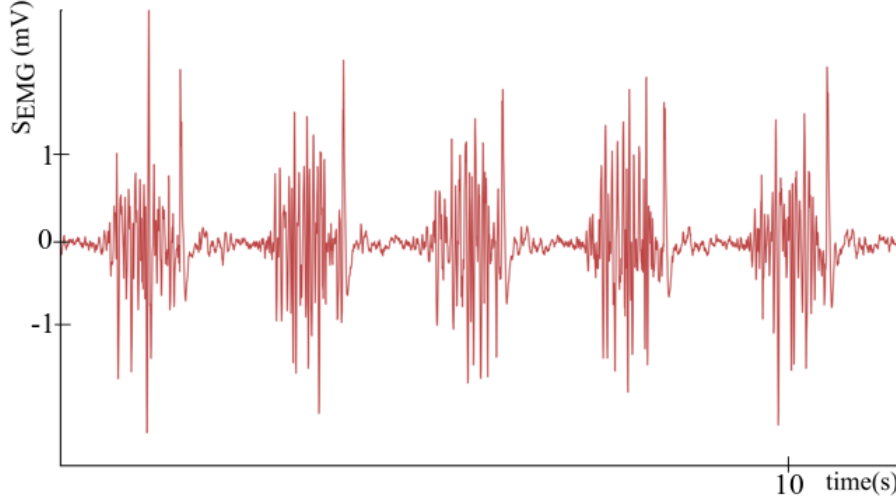


Figure 2.10: Example of an EMG acquired from a hamstring during a cycling exercise.

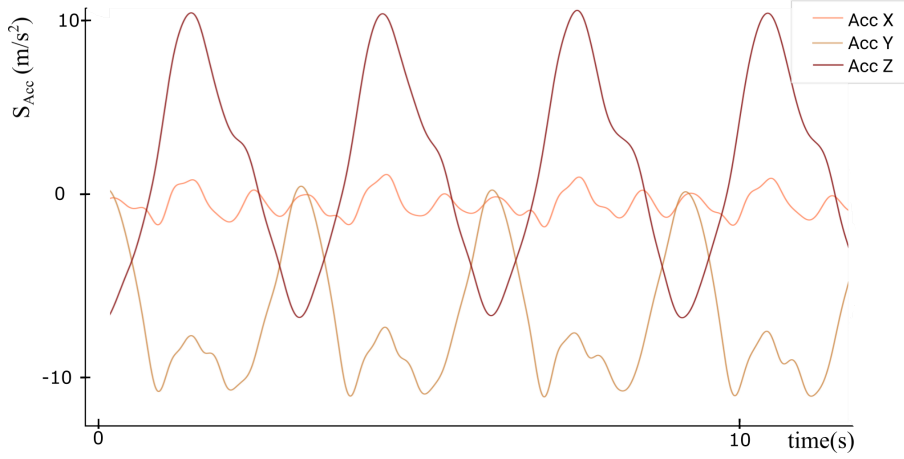


Figure 2.11: Example of an ACC acquired from a hamstring during a cycling exercise. The cyclic motions are the representation of the repetitive movements of the leg in the three axis of motion.

that are used to move the eyes.

2.5.1.2 Accelerometer

The Accelerometer (ACC)s are sensors that capture motion by detecting changes in the acceleration in three orthogonal axes (Fig. 2.11). These sensors are inertial-frame, therefore the values obtained are a differential to the Earth gravity values, obtaining 0 m/s^2 when the device is falling and 9.8 m/s^2 when standing still. The static component is the Earth's gravity while the dynamic is the movement performed by the area of contact [38]. It is possible to isolate components of an ACC when subjected to filters, such as the linear acceleration of a device and the gravity components with low and high-pass filters, respectively [39].

Neuron

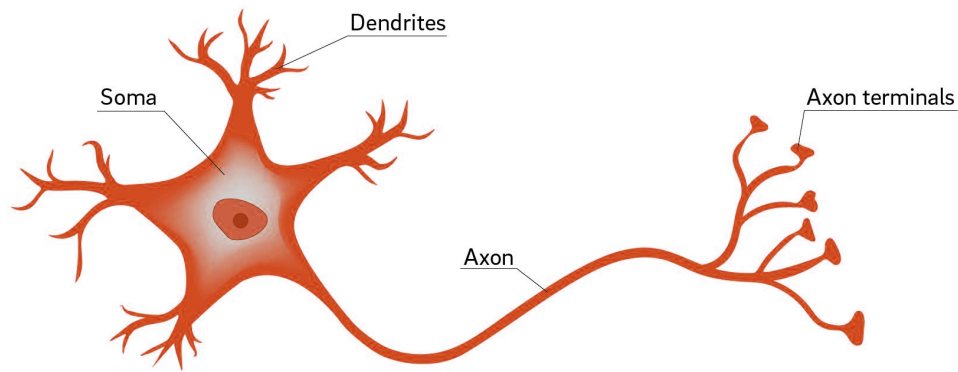


Figure 2.12: Diagram of a neuron.

2.6 Nervous System

The nervous and endocrine systems are responsible for the control of the homeostasis of the human body. The former is comprised of the sensory input, the processing system, and the motor output. The sensors receive information from the body state or the surrounding environment. The processing system, i.e. the central nervous system, is composed of the brain and spinal cord. The brain is both a storage and generator of information, while also being capable of reacting to a stimulus. Both input (e.g. sensory data) and output (e.g. muscle contraction) pathways are carried through nervous cells by the spinal cord and peripheral circuits. The autonomous nervous system is the portion that operates at a subconscious level and controls many functions of the internal organs, such as heart activity, secretion of body glands and gastrointestinal tract contractions [24].

2.6.1 Neuron Physiology and Synapses

The functional unit of the Central Nervous System (CNS) is the neuron, as depicted in Fig. 2.12. More than 100 billion interconnect with each other through the dendrites. These structures enlarge the cell surface by branching where the nerve impulses, the synapses, are received. These action potentials normally pass only in the forward direction through the nucleus, axon, and the terminal area, before being transmitted to the dendrites of the next nerve cell. This feedforward mechanism forces the signal to travel in one specific direction to perform specific functions [24, 40].

The synapses, or nerve impulses, are information that is transmitted in the CNS and can have different behaviors:

- may be blocked during the transmission to the next cell;
- may be changed from an impulse to a succession of impulses;
- several impulses may add with other impulses from different sources.

2.6.1.1 Synapses

Synapses can be chemical or electrical, and most of the signal transmission between neurons is chemical. The nerve ending launch neural transmitters that stimulates the protein gates of membranes of the next neuron (Fig. 2.13). Some of the best-known chemicals are acetylcholine, norepinephrine, histamine, and serotonin. This stimulus may be excitation, inhibition or sensitivity. These logistics are desirable as it permits the synapse to travel in only one direction. In opposition to the electrical synapses, which can travel in both directions. These last are characterized as directly opening of the channels that conduit electricity [24].

The one-way control mechanism permits the complex and subtleties required for highly specialized cells and centers within the nervous system. This specialization relies mostly on the difference in the anatomy of the neuron. The size of the cell body, properties of the dendrites (size, shape, and number), length and size of the axon and the number of presynaptic terminals, alter the reaction to the synaptic signals, and, therefore performing different actions [24].

2.6.1.2 Presynaptic Synapses

The presynaptic terminals lie in the surfaces of the dendrites and soma (main body of the cell) of the neuron. These are small oval knobs that comprise two internal structures: the transmitter vesicles that contain the substances that either excites or inhibits the synapse, and the mitochondria that provide the energy required to synthesize new transmitter substances. The action potential induces the presynaptic terminal to release these to the medium. This action potential depolarizes the membrane opening the channels required creating the ionic differential on both sides. Once again, the vesicles are recycled and the active transport reestablishes the resting position [24].

2.6.1.3 Postsynaptic Synapses

The value of the resting potential of the neuron dictates its excitability level. As the value is more negative, the less excitable it is. For example, the spinal motor neuron has -65 mV is more susceptible than a neuron in the skeletal fibers that have -90 mV.

The spinal motor neuron can be triggered when reaching the value of -45 mV, but for the differential to reach the +20 mV, about 40 to 80 anterior terminals need to discharge synchronously. There is also the possibility of inhibitory events, caused by the opening of anion channels, causing the resting position more negative, reducing the sensitivity of the neuron [24].

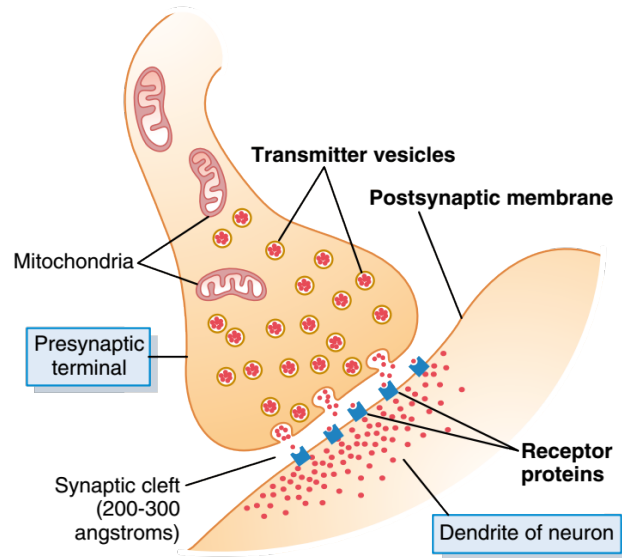


Figure 2.13: Synaptic release of the transmitter proteins, from [24].

The excitatory postsynaptic potential only is created, in the initial part of the axon, if the discharges add to the positive. This added potential (synaptic summation) has a window from 1 to 2 milliseconds and slowly decreases during 15 milliseconds until the resting potential. There are two types of summation, the temporal (the higher the synchronicity, the higher the value) and spacial (all soma has the same potential). Sometimes the excitation is not enough for the excitation, but the neuron still fires, it is said that it was facilitated [24].

2.6.2 Information Processing

Not all the sensory information is used, as only 1% is deemed relevant or important. Examples are the sense of touch from wearing clothes or even hearing its own heartbeat. Rare cases of important sensory information are immediately channeled into the relative integrative areas where it is processed [24].

These attentive mechanisms are due to the differential spread of synapses in the cortex pathways. Some pathways exhibit fast and easy and fast communication, while others block the connection. Facilitatory and inhibitory signals can also control this synaptic transmission. The number of impulses also adds to the postsynaptic response to occur, but sometimes some neurons amplify weak signals channeling in many directions instead of only one [24]. These complex mechanisms and pathways are controlled by the complex system of the CNS.

2.6.3 Central Nervous System

Due to evolutionary development, the human nervous system has acquired three functional areas: spinal cord, subcortical, and cortical. Apart from receiving stimulus from

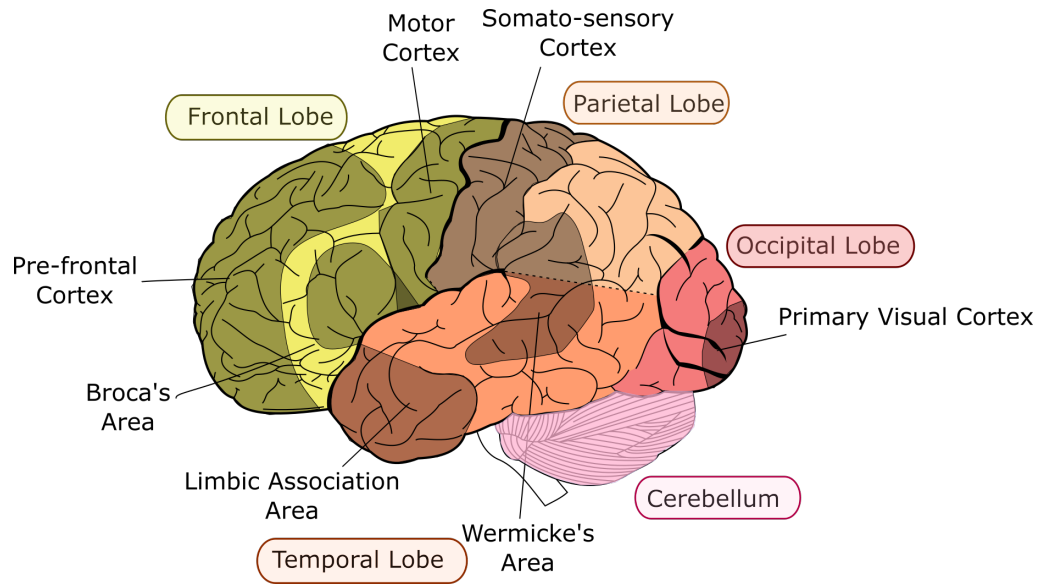


Figure 2.14: Brain areas of the brain.

the peripheral nerves and brain controls of the body, the spinal cord is also responsible for several functions such as walking movements, reflexes, local control of blood vessels, gastrointestinal movements, urinary secretion. The cortical area sometimes even operate by sending signals for the cord to operate autonomously. The subcortical area is composed of the lower brain areas in the medulla, pons, mesencephalon, hypothalamus, thalamus, cerebellum and basal ganglia. This area controls most of the subconscious activities of the body such as control of the arterial pressure, respiration and salivation reflexes. Some emotional patterns are also correlated with this area. The cortical area is responsible for the cognitive functions, such as memory, thought processes and information perception. It only functions in association with the other CNS areas [24].

2.6.4 Brain Cortex structure

The brain cortex may be segmented into four distinct lobes as depicted in Fig. 2.14: occipital; parietal; temporal, and; frontal lobes. The complex processing of the inputs from the primary sensory cortices (part of the parietal and occipital lobes) and the higher cognition function of the brain is the function of the association cortices (part of the parietal, temporal and frontal lobes). The parts of the brain responsible for encoding sensory information and commanding movements occupy about a fifth of the cerebral cortex, where the rest is to identify relevant features of a stimulus, object recognition and planning the appropriate responses, including storing aspects of the information. This ability is called cognition and is a complex task that requires the association cortices to receive and integrate information from a variety of sources in several cortical and subcortical areas [41].

The circuits involved in the information transmission in the cortex are made through

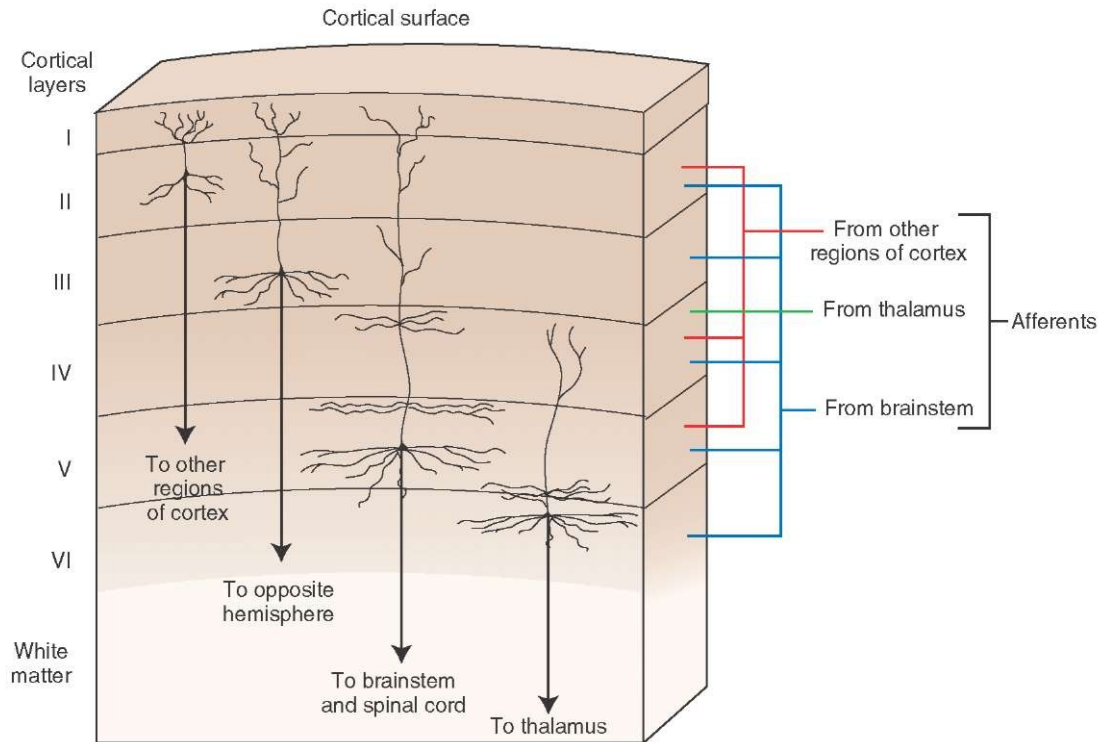


Figure 2.15: Histological appearance of the motor cortex. From [42]

six layers (Fig. 2.15). Each layer has different properties according to the corresponding cell's sizes, densities, shapes, inputs, and outputs. Each lobe also comprises numerous subdivisions, named cytoarchitectonic areas, that connect with each other by local or long-distance connections. These circuits have similar properties [41]:

- each one has a preferential source for input and a primary target for the output;
- There are vertical and horizontal connections;
- cells with similar functions, throughout all the cortical layers, tend to be arranged sequentially in radially aligned groups;
- Some neurons (interneurons) have more extensive axons that connect horizontally to other functionally similar groups of cells.

Even though these similarities are common in most of the cortex, connectivity between the association cortices is different from primary and secondary sensory and motor cortices. They can receive information from other sources and can also have higher enervation in several direct connections (corticocortical connections) [41]. These pathways are the source of the mechanisms used in higher cognitive tasks such as processing visual and memory information.

2.6.5 Sensory Processing

The sensory receptors of the human body, such as the visual receptors in the eyes, auditory receptors in the ears, tactile receptors in the surface of the body, taste receptors in the tongue, between others make us aware of our surroundings. The somatic portion of the sensory system, when excited, transmits information from deep structures and enters the CNS through the peripheral nerves. The spinal cord and several areas of the brain, such as the thalamus and cortex, receive this data to be further processed. The output of the brain can be stored for future use or it may also trigger an immediate reaction through the motor areas of the nervous system [24, 41].

These include effectors which are the muscles and the glands that secrete active chemical substances. The autonomic nervous system, controls the smooth muscles, glands, and other bodily systems, while the skeletal muscles can be controlled by the CNS. Lower areas of the CNS play an instantaneous and automatic muscle response to the stimuli, while the higher are more involved with perception, processing, and analysis of what should be learned. All of them contribute in different parts to the permanent memory storage [24].

2.6.6 Memory Processing

Most of the filtered sensory information is stored to be used for thinking and for later motor activities. The spinal cord, the basal regions of the brain and, with a higher percentage, the cerebral cortex store information. This stored information is a function of the synapses, such as, when a pathway is constructed due to the passage of a large number of times of a specific sensory signal, that pathway becomes facilitated [24]. The brain can also cause these synapses to occur, even when the sensor is not triggered. Therefore some sensations that can be experienced only with the memory recall [24].

Even though the long-term facilitation is not a known process, but once memories are stored, they become part of the processing mechanism for the future thinking process. As the brain compares new sensory experiences with old, these help to create new pathways to store important new information [24].

The hippocampus and the medial temporal region have several connections with the cerebral cortex, namely in the areas responsible for thinking and language. Working memory is a type of buffer that enables the brain to retain memory over some time, just enough to give an answer, such as waiting for a sentence to end to understand the meaning. This function depends on the prefrontal cortex.

Language highly depends on memory. Even though these mechanisms are not fully understood, research indicated that the underlying structure that promotes speech, located in the left hemisphere at the temporal lobe level is close to the Broca's area in the frontal lobe, where the vocal expression is planned. This signal then travels to the nearby area of the motor cortex activating the required limbs for phonetics. In the level of reading, when a word is read, the information is transmitted from the primary visual cortex to a structure, the angular gyrus, where is matched to the sounds of the words that are spoken.

Then this word passes to the auditory area, where it can be perceived. For writing the pathway comes from the oral instruction from the frontal lobe and passes through the same pathway ending in the motor area [41].

Research tend to map the learning and memory mechanisms in the brain, as these processes tend to be stored in different parts of the brain, depending on the type of information. These types can be classified as procedure knowledge (learned habits) and declarative knowledge (familiar experiences). An important factor that influences how a piece of information is stored, depends on the rewarded or punished consequence. This establishes the behaviors that the human will learn and remember. The amygdala has an important role, as it provides the emotional trigger for these scores [41].

2.6.7 Visual Sensing and Processing

Smooth muscles and the parasympathetic nervous system control the eye to ensure that the input of the light gives a good quality of an image. The eyes contain different cones that are sensitive to different colors of light. The tricolor mechanism of color detection is that the eye can detect all gradations of red, green and blue (RGB) monochromatic lights. These provide several possible combinations, based on the percentage of excitability of each cone, giving us the feeling of color.

The contrast of the image is made by horizontal cells that inhibit the electrical signal through the several layers of cells. Therefore, the transmission of the proper visual patterns does not spread light intensity to other parts of the information regarding the image.

The nucleus of the thalamus, located at the dorsal end of the thalamus serves two principal functions: preprocess information from the optic tract to the visual cortex, and; filter the information given to the cortex. According to the first function, it transmits high fidelity information, due to its proximity to the retina. The information from both eyes is kept apart and each pathway contains six nuclear layers that receive information from different parts of the retina.

The occipital lobe is responsible for the perception of almost all aspects of visual form. The visual cortex is divided into primary and secondary.

The first is located in the calcine fissure area, extending to the medial area of each lobe. This area has direct access to the visual signals, where each area is correspondent to a part of the image from the retina. Like all the cortex, it contains six layers, where each deeper layer conveys processed information from the top layers. Each layer also passes horizontal information to other processing areas, namely the surface of the cortex, or adjacent color detecting units (which are spread across the visual cortex). The information from both eyes is only interlaced in the fourth layer of the primary visual cortex. After this processing, the information proceeds to two major pathways before reaching the secondary visual cortex: the position and motion and, the accurate color pathways.

The secondary visual cortex is located in the surrounding areas of the occipital lobe and analyzes meanings from the images. The various subareas analyze and scrutinize

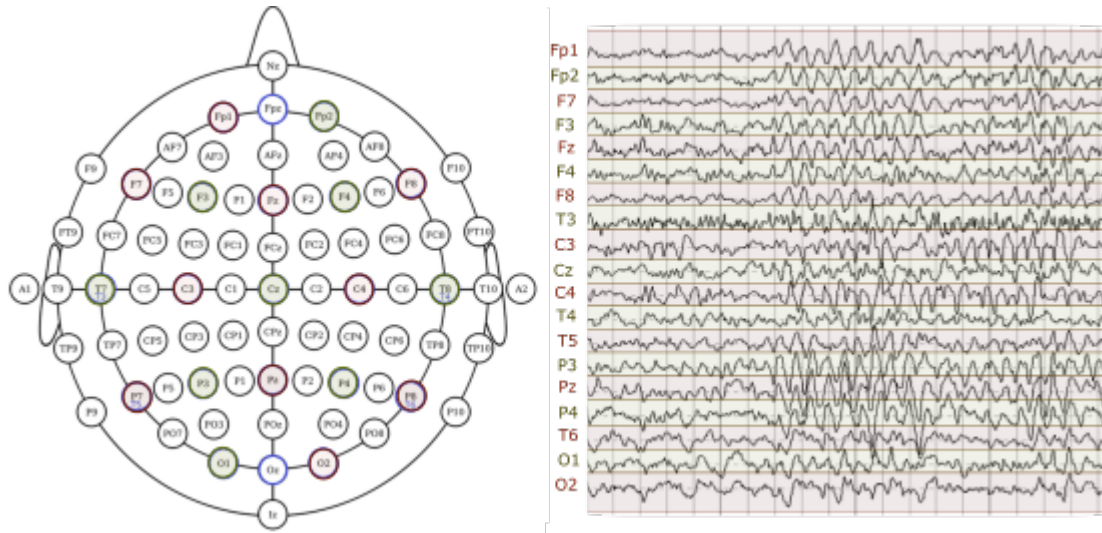


Figure 2.16: Example of an acquisition of an EEG. In the left EEG electrode positions based on the 64-channel of the international 10-20 system while in the left the acquired signal from 19 sample electrodes. From each channel, the A1 and A2 were subtracted concerning the opposite hemisphere. E.g. A2 was subtracted from Fp1 raw signal, whereas for Fp2, A1 was removed.

progressively. There are specialized areas in detecting borders, where the image change from light to dark areas. They trigger with more intensity when sharpness is higher. The orientation of lines is also functions that some areas of the visual cortex possess, differentiating angles stimulate and inhibit specific neurons, increasing the contrast edge and gives the orientation perception of the line. These neurons are simpler and have a different composition from the normal neuronal cells.

To understand an object several cells that detect angles, lines or a specific length, angular shapes, or other characteristics are progressively triggered. This simpler information together brings higher orders of information from the visual scene that enables the perception of what is being observed. In terms of color the same happens, but between different tones of color and intensities. Different neurons fire when subjected to different contrasts. The light intensity such as brightness is also a feature that is extracted through the edge filters that are learned in image processing. These are responsible for detecting where an object end.

2.7 Central Nervous System biosignals

The EEG (Fig. 2.16) acquires the spontaneous electrical activity of the brain by using multiple scalp electrodes. The differences in potential are seen due to the equalizing currents of the extracellular space. Due to this effect, the firing of many neurons is captured in each sensor, and the higher the synchronicity of their firing the higher the amplitude seen in this biosignal. There are different configuration setups for electrode

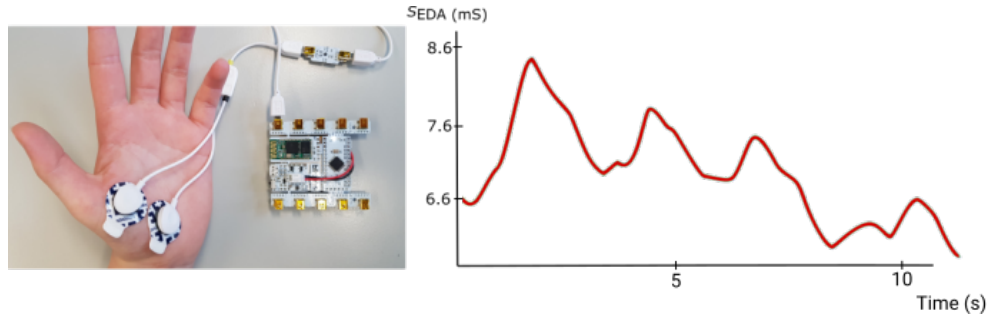


Figure 2.17: Example of a acquisition of an system for EDA. In the left the Bitalino setup for EDA from [45], and in the left a graph of on acquired EDA from [46]

placement, that range from two to 256 array of electrodes. Each point is located in a specific named location, with a letter for the brain region and a number that indicates the radial position (except for z that indicates the central zone) [14].

Several problems add resistance to the signal, such as the scalp skin, dead cells in the surface and hair. Even though they have an apparent random behavior, different rhythms in different areas of the brain indicate different stages of activity. For example, the waves between the range of 8-13 Hz (alpha waves) are related to relaxation, but also to when the eyes are closed, or when performing a higher cognitive task. The beta waves, between 13-30 Hz are related to an active state. The theta waves indicate drowsiness or states of meditation, while the delta waves are when someone is in deep sleep. The gamma waves are waves with higher frequency than the beta waves (25-100 Hz) and can be seen during alertness after sensory stimulation [14, 17].

The Electrocorticogram (ECoG) is a mesh of electrodes that are placed directly over the cortex, while the Microelectrodes (ME) are electrodes placed directly on the cortex. These are considered the gold-standard technology on recording action potential directly. As the recording from both these devices are spikes of potentials that each specific neuron fires. These technologies pose many issues, such as the material must offer biocompatibility, safety, and cognitively. This also requires a very invasive method of insertion, and therefore not many patients have used them. Other new electrodes, called Thread Electrodes (TE) were developed by Neuralink and the technology is also invasive, but each electrode is significantly smaller (in the order of micrometers) and the materials are more flexible [43].

2.8 Electrodermal Activity

The Electrodermal Activity (EDA) is an index of the sympathetic nervous system activity (Fig. 2.17). The electrical properties of the skin change according to our emotional state. The EDA measures the conductivity of the skin between two points of contact and measures the current flow between them. This biosignal includes a background tonic, slow changes related specifics of the individual, and rapid phasic components, related to changes in emotional or stress states, that result from the sympathetic neural activity [44].

EDA is widely used for detecting cognitive states, arousal, changes in emotion, and attention. It is also used to examine unconscious responses such as threat and anticipation or even in the detection of personality traits, based on given stimulus and included in a mixture of other signals [44].

Deep Neural Networks

The area of Artificial Intelligence (AI) embraced Deep Neural Networks (DNN) has a powerful technology that achieves results that surpass many of the previous Machine Learning (ML) algorithms. But the implementation of Artificial Neural Networks (ANN) has been hindered throughout the years due to its complex nature that required the development of capable training algorithms and faster computing hardware. This chapter will start by delivering a brief description of the story of AI, in particular of ANN, and later the explanation of the basics of ML and DNN. These contents include the definition of supervised and unsupervised learning, of the atomic structure of the neural network and their activation functions, and the DNN building blocks that result from the combination of the previous. At last, it will be given a summary of common architectural paradigms on how to optimize their parameters through training.

3.1 History of Artificial Intelligence

In 1948, Alan Turing submitted a technical report called "Unorganized Machines" in which he presented his ideas of creating machines that could have intelligent behavior and debuted the concept of connectionism, suggesting that randomly organized networks of neuron-like nodes could be used for computation [47, 48]. The concept of "Unorganized Machine" is compared to the cortex of a human infant, which could be trained to readjust the network connections to turn the machine capable of performing specific tasks [47]. These concepts were created before the era of digital computing.

During 1956 a summer workshop, with the title of Dartmouth Summer Research Project on Artificial Intelligence, was widely recognized as being the founding event of AI field. Only in the 1960s, AI started taking its form, where programs were written following logical rules governed by a careful representation of previous human knowledge. Even if

simulated, to a certain degree, human decision-making algorithms did not learn. Some expert-knowledge systems were made but there was no flexibility using if-then-else rules. When the funding parties realized that AI was going to take longer than expected to achieve its goals, research dropped from mid 70s until the start of the 80s. This period was known as the first "AI winter".

The second rise started with the adoption of data-driven powered by statistical ML methods in the area of speech and image recognition, against the knowledge-based recognition. Therefore, the engineers did not need to construct precise and exact rules, as the new algorithms focused on statistical models or simple ANN. The created machines are tuned with parameters that can be learned or regulated by generalizing the information contained in data. By making an abstract view of some aspects of the problem, these machines are capable of dealing with uncertainty in a specific context. Examples of these algorithms are the Bayesian networks, support-vector machines, decision trees and, now called shallow Neural Networks (NN). Some successful works were made in perception-speech, face, visual objects and handwriting recognition [12].

The two major limitations that made ANN research hindered during the next 20 years were the vanishing gradient and the high computational complexity. The former is reported when the layers near the output are prone to learn faster and retain information, while the ones near the input did not learn. The learning mechanism was not able to share information as the networks became deeper, causing the gradients close to the input close to zero. As for the required high-level computation, even though the first DNN was created during the '80s by LeCun et al. (1989) [49], it was not popular as the required computational requirements were too high for the average computational systems.

During that time, Hidden Markov Models (HMM) were the most used generative model for speech recognition, which were integrated with Gaussian mixture models. Some versions of this algorithm included expectation-maximization, extended Kalman filters, and backpropagation for learning the parameters and used a multiple latent layers representations for the generative process of speech waveforms that could be transduced to a human-speech synthesis, giving birth to the first commercial success of deep structures [12].

This generation of AI algorithms outperformed the previous one, but they were far from human-level performance. The shallow ML models were not able to absorb all the information contained in large datasets and the learning algorithms and computing infrastructures were still not powerful enough. Due to the underestimation of these problems, the second "AI winter started" on the late 1980s.

During the 2010s the wall was broken with the adaptation of parallel programming capabilities of the Graphics Processing Unit (GPU) to train deeper NN and solving the problem of the vanishing gradient, which interfered with learning the parameters with higher amount of data [12].

The third rise of AI was introduced by the breakthrough of deep learning that encompasses a higher amount of layers that can assimilate more information augmenting the

abstraction capabilities. As the traditional ML capabilities were based on human-designed features relying heavily on human expertise. DNN learns and perceives abstractions by learning new features in their layered model structure in the training phase, taking advantage of a high amount of data. Therefore DNN is one of the main reasons for the resurgence of ANN. From 2009 several industrial applications of this technology were deployed, being launched on large scale and embracing both industry and academia [12].

Nowadays a large number of real-world applications may be found in several industries, such as image captioning, visual question answering, web search, natural language processing, drug discovery and toxicology, customer relationship management, recommendation systems, medical informatics, Internet advertisements, medical image analysis, robotics, self-driving vehicles, board games among others [12].

3.2 Types of Machine Learning Methods

Before diving into neural networks one should understand the concept of supervised and unsupervised learning. Supervised learning is when the model that we wish to train has a ground truth to compare the predictions, i. e., if each record has an associated label the model's parameterization is based on the match between the input and the predicted output. In this segment, there are two main types of prediction, regression methods, which the model is approximated to a function that computes a $f(x)$ for each value of x , whereas the classification method gives the probability of an output based on a probability mixture, where the most probable output is a class. Therefore, the mentioned matches the observed values to the probability of the output being of the class k knowing the input x , i.e. $P(y = k|x)$. The classification will be the one with the highest likelihood.

Supervised methods rely on the labeling of data, which is not always possible due to the massive quantity of data and requirements to have specialized humans or high-reliable algorithms that give sense to the given information, which can be both time-consuming and economically costly.

The unsupervised methods do not need ground truth, as it couples proximal feature distributions. Typically methods such as clustering, calculate the similarities and join them in groups, also known as clusters, giving similar traits to regions. Architectures, such as autoencoders (explained in section 7.2), can be unsupervised when they only need information from the input to be trained and implemented.

3.3 Single Layer Perception

A single-layer perceptron is depicted in Fig. 3.1 and is the atomic component of the NN as it simulates mathematically a simplified approach of how a single neuron works [50]. Therefore this structure is the key component for multi-layered networks and it exhibits behavior that enables the study of non-linear characteristics of models such as the multilayer networks [51].

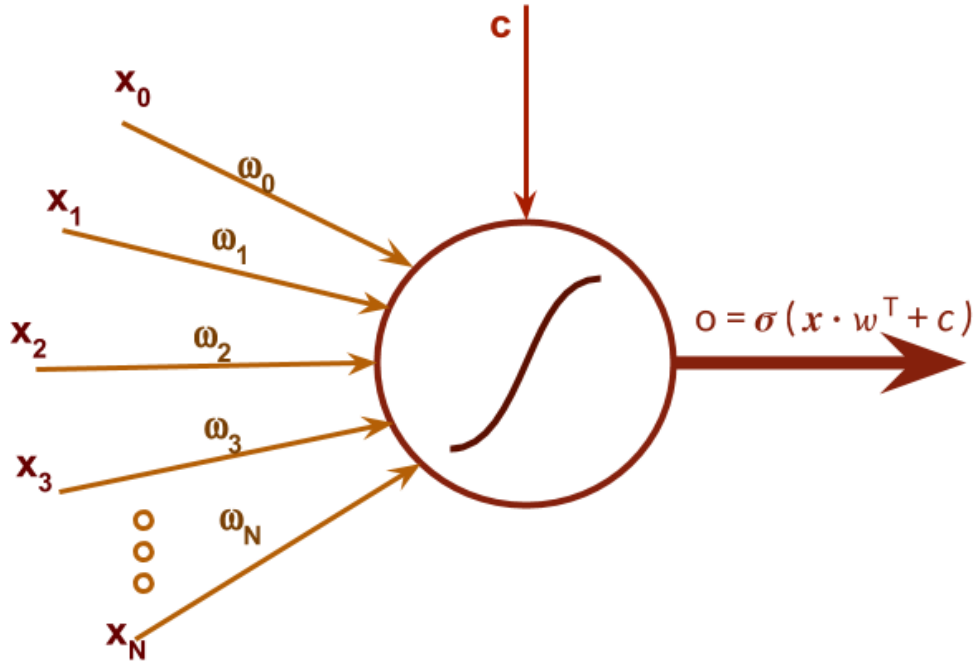


Figure 3.1: Artificial single neuron representation.

The input channels, represented by x_n , are the components that substitute the dendrites. Each weight symbolizes the facilitation or inhibition properties that the pathways possess when transmitting the synapses. Just as the soma, the neuron sums the various input signals and forms an action potential as soon as a threshold, in this case, the bias value (c), is surpassed. The activation function dictates the strength of this activation. The neuron represented in Fig. 3.1 has the sigmoid function that behaves like a switch, achieving only the values of 0 or 1.

3.4 Activation Functions

Historically the activation function was just an "if" operation, where the if result value is 0 then its negative, while 1 if positive. After the creation of the backpropagation method, which requires the computing of the derivative of the weights, the function had to change for the derivative to be finite when at the input of zero, and thus the creation of the sigmoid function. Fig. 3.2 depicts examples of activation functions used throughout this thesis. Each one will be explained briefly.

3.4.1 Sigmoid and Hyperbolic Tangent

The sigmoid function (Fig. 3.2 (b)) imitates the behavior of the switch statement, but its continuity allows the calculation of the derivative. The graphic of the hyperbolic tangent

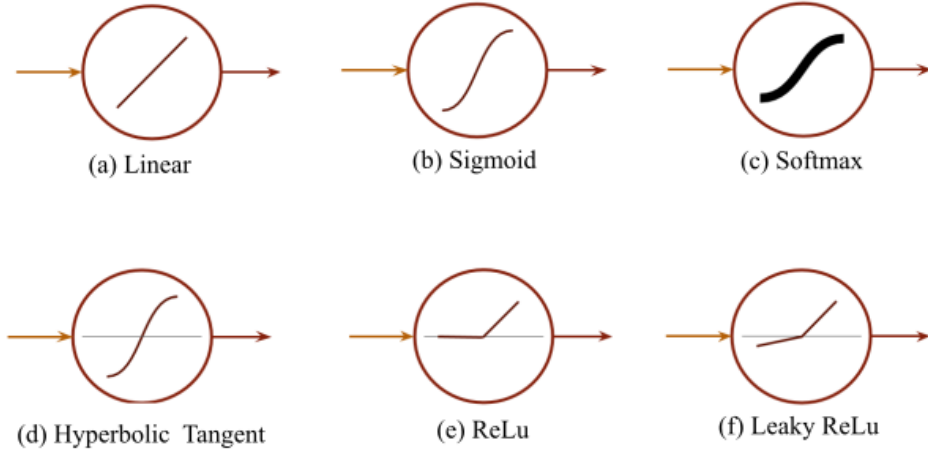


Figure 3.2: Activation functions used in DNN. The style of the softmax activation makes the differentiation from the sigmoid function.

(Fig. 3.2 (d)), abbreviated as *tanh*, has a similar shape, but the values of activation are different. As the sigmoid functions, the values are comprehended as 0 or 1, the tanh change to -1 or 1. Their correspondent equations are as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.1)$$

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (3.2)$$

These activation functions are normally used in Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) for memory control (see section 3.5.3) or for normalizing the output of a layer.

3.4.2 Rectified Linear Unit (ReLU) and Leaky ReLU

Even though the ReLU function (Fig. 3.2 (e)) is not continuous, this activation function has been very successful while using Convolution Neural Networks (CNN), as it reduces the effect of the vanishing gradient. The ReLU function is represented as $\max(0, x)$ has a minimum value of 0 and it doesn't have a maximum value. Several derivations of this rectifier unit exist, such as the Leaky ReLU (Fig. 3.2 (f)), *Softplus* and noisy ReLU.

3.4.3 Linear and Softmax

The NN may also include linear functions (Fig. 3.2 (a)), mainly for the classification layers. Usually, it makes the connection between the fully connected and the *Softmax* layers, as the output is generally named logits has the moment before converting into a probability function. The *Softmax* (Fig. 3.2 (c)) is the transformation made for classification problems,

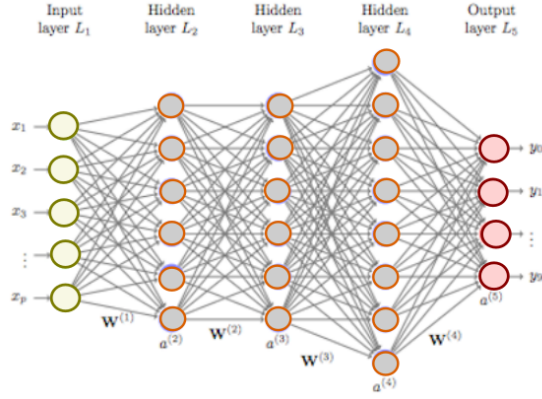


Figure 3.3: Diagram of a fully connected network

where the output vector has a total sum of 1. Each position will indicate what is the probability of the class in the coded position. The equation is as follows:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ for } i = 1, \dots, K \text{ and } z = (z_1, \dots, z_K) \in \mathbb{R}^K \quad (3.3)$$

3.5 Deep Neural Networks

DNN are more complex forms of NN as they have a higher number of layers that learn non-linear problems and store more information, therefore they need more information to learn and generalize problems and functions. Two main architectures arise from this paradigm: CNN and Recurrent Neural Networks (RNN). The first relies on a steady-state space, where it learns the spacial representation of a steady-state data, whereas the second is a state machine that learns the sequential pattern of states in which the current depends on the previous one. Both of them will be presented in more detail.

3.5.1 Fully Connected Network

A fully connected network (Fig. 3.3), also called dense network, is a combination of several simple perceptions (nodes), where all the nodes are interconnected with the previous and the next layers. As the input layer receives the first influx of information, the output transmits the transformation made by succeeding simple mathematical combinations made by the hidden layers, i.e. the in-between layers. Together they formulate complex non-linear calculus that ends in an activation function. The extra hidden nodes and successive layers increase the memory of the network, but this also increments the difficulty for training, and thus, the probability of vanishing the gradient.

These networks can be used for two main purposes: interconnecting incompatible layers (make connections between two layers with different dimensions), and make relations between the extracted features and the output (classification and regression in the last layers).

3.5.2 Convolutional Neural Network

The convolutional neural networks are a type of neural networks which applies convolution operations between the weights (called kernels or filters) and a tensor. In the case of image recognition, the image can either be provided as input to a CNN in the form of a 2-dimensional tensor (width and height), in case of having a single value that represents each pixel (grayscale) or as a 3-dimensional tensor (width, height, and number of channels), where the last dimension represents the components of each pixel. For example, the intensities of red, green and blue are the three channels that can be used for RGB images.

The kernels can be interpreted as filters that detect shapes, edges and other patterns that may appear in the image. These are composed of weights that can mutate as they learn, during the training process. In the forward propagation step, kernels are activated by passing through an activation function, a nonlinear function, such as the hyperbolic tangent (\tanh) or the ReLU. The latter is given by $\max(x, 0)$, being x the input vector. It passes only values above zero, resembling its biological counterpart, the action potential. This feature allows the NN to solve nonlinear problems.

All the layers in a CNN that operate by convoluting kernels and tensors are called convolutional layers and their output is the stacking of two-dimensional feature maps. After a convolution, a pooling layer can be used to reduce the dimensions of the feature maps. This procedure does not only minimizes the computational effort, but also helps filters in the detection of shapes that are submitted to translations [52]. In this task, this operation consists of replacing consecutive patches of size $n \times n$ by either their maximum value (max pooling) or their average value (average pooling).

After the inputs are passed through a sequence of convolutional and pooling layers, to make it possible for the classification step to take place, the final outputs are flattened and can pass by one or more fully connected (regular) layers, until the class with the highest probability is selected as the predicted class. The fully connected layers are simply composed of matrix multiplications between the inputs and the weights, with an optional addition of a bias parameter.

3.5.3 Recurrent Neural Networks

Typically, CNN accept and generate fixed-size vectors or matrices as inputs, without acknowledging the sequence dependencies between two adjacent states [23]. On the other hand, RNN are an extension of a conventional feedforward neural network, in which feedback loops are enabled (Fig. 3.5). These loops allow the coherence between two succeeding states, by transmitting a vector of activations between them. As new information is received and compared to the previous cell state, it will observe the dynamic changes over the sequence [54, 55].

The recursive nature of these structures allows each cell to maintain or update its hidden state, based on the received input and the last state. This hidden state is the understanding that the network has of inner the generation of the observable sequence.

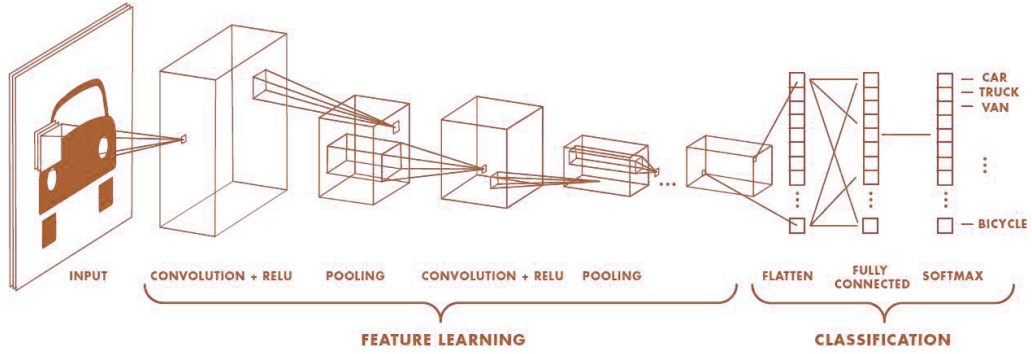


Figure 3.4: Diagram of a example of a CNN for object recognition, from [53]. As the car image is convoluted by a series of windows and an activation function is made a pooling step reduces the dimensional of the resultant tensor. As this process is repeated the features will be extracted. In the end, the fully connected layers and *Softmax* will make a classification based on the information extracted before. Each position of the output vector will have a probability value associated to a class.

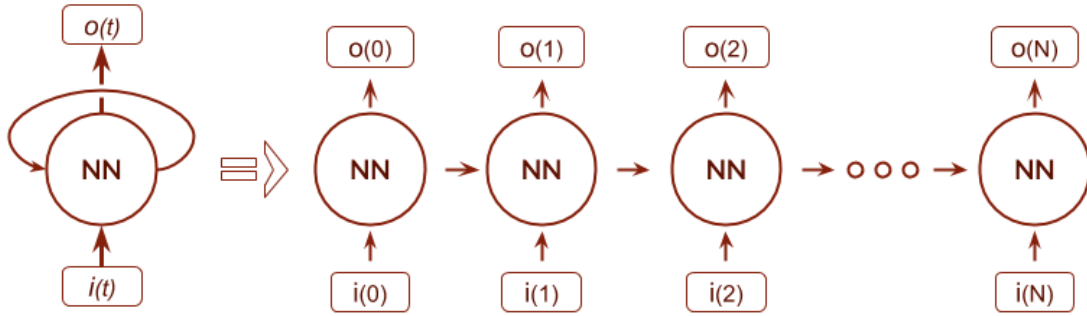


Figure 3.5: Unfolded RNN.

The nature of these networks make them deep and adapted in working with sequential data [23, 56].

There are several arrangements on how to use RNN depending on the problem to solve. Fig. 3.6 portrays three types of configuration. The "many to one" configuration (Fig. 3.6 (a)) is when the information between all time-steps is not relevant, and only the whole combination of sequence matters. This paradigm is most frequently used in classification problems. When the cell outputs occur in more than one state than we are dealing with the "many to many" setup. There are two types that can be discriminated, the first is the encoder-decoder configuration (Fig. 3.6 (b)), where the input sequence gives the detail of the whole sequence in the last encoding state, which is then decoded in a series of outputs. This mechanism is used in translation, for example, where the input reflects one language, while the translated output depends on all previous encoding. The second is when each time-step has an output (Fig. 3.6 (c)). This is especially favorable for prediction devices when it is wished to predict the next value based on the current one.

One of the great challenges faced in the early stages of RNN was how simple it was

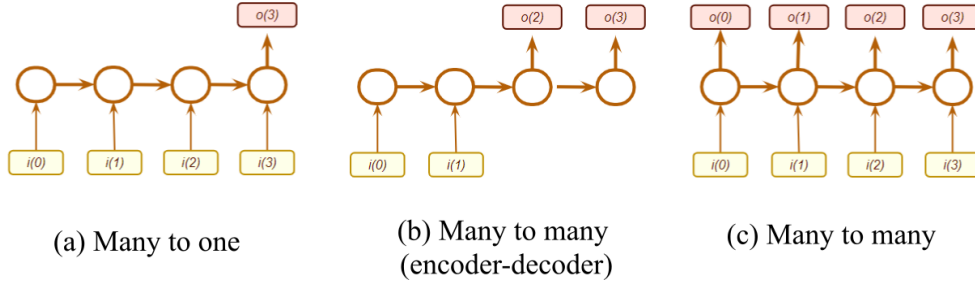


Figure 3.6: Three different configurations for RNN.

for the gradient to vanish once the sequence was too large. The increased depth made the computations of the long-time dependencies difficult or even impossible. In order to disentangle the weights during the learning process, a method was introduced that allowed the network to forget, which in turn allowed it to absorb higher magnitudes of knowledge [23, 57]. These networks have two base implementations named LSTM and, more recently, the GRU.

3.5.4 Long-Short Term Memory

A basic RNN merges the old state vector with the new inputs in one layer, whereas LSTM have typically three gate layers that protect and control the cell state, providing the LSTM with the ability to continuously write, read and reset operations for the cell state [58].

The first layer is involved in managing the information from the input that can be ignored and is controlled by the forget gate. The second layer, which contains the input gate, handles which information should be neglected. Both gates are responsible for updating the cell state based on its previous seen states and the new inputs. The final gate filters the values that will be fed into the gates of the next state. The mathematical background of the forget (f), input (i) and output (o) gates for each state n are reflected in the following equations:

$$f_n = \sigma([W_f x_n] + [U_f h_{n-1}] + b_f) \quad (3.4)$$

$$i_n = \sigma([W_i x_n] + [U_i h_{n-1}] + b_i) \quad (3.5)$$

$$o_n = \sigma([W_o x_n] + [U_o h_{n-1}] + b_o) \quad (3.6)$$

where W matrices are the weights related with the input x_n and U are the weights related to the last seen state (h_{n-1}), and both groups of weight matrices are associated with their own specific gates (indexes f , i or o). All the resultant products are mediated by bias (b).

The candidate for the next state (\tilde{C}) and the next state is a product of:

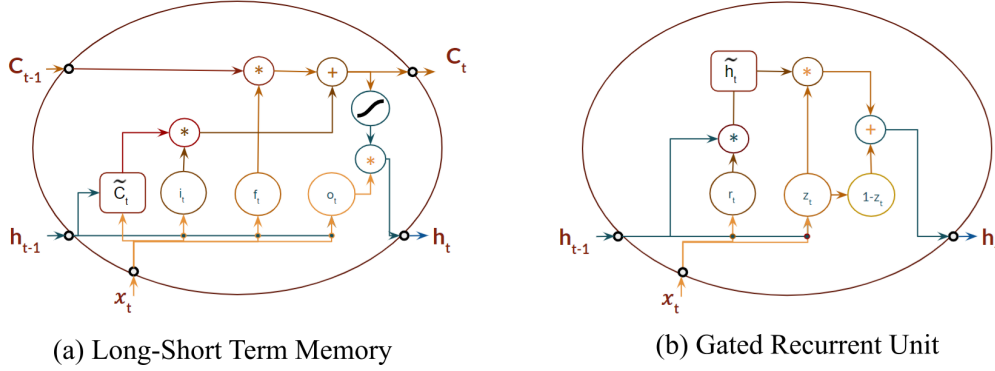


Figure 3.7: Single unit representation of the LSTM (a) and a GRU (b) NN

$$\tilde{C}_n = \tanh([W_c x_n] + [U_c h_{n-1}] + b_n) \quad (3.7)$$

where the W_c and U_c are matrices that transform the input and last output, respectively. The \tanh modulates the output of the function between -1 and 1. Finally, the hidden state (C) and the seen state (h) are results of the following equations:

$$C_n = \sigma(f_n \cdot C_{n-1} + i_n \cdot \tilde{C}_n) \quad (3.8)$$

$$C_n = \tanh(C_n \cdot o_n) \quad (3.9)$$

as seen from these equations, C_n at this timestamp depends on what information the forget gate lets pass from the previous hidden state C_{n-1} while the input gate controls which information of the \tilde{C}_{n-1} passes. At last o_n filters what information the gates of the next state will have access to [58].

3.5.5 Gated Recurrent Unit

The LSTM and GRU manage memory in different ways. While LSTM has several separate memory units, GRU comprehends these in two main gates – the update gate and the reset gate. This makes of GRU a simplification of the LSTM, in which only two gates are responsible for a more efficient management of memory [59]. Fig. 3.7 represents the different steps that constitute the dynamics of the GRU.

The reset gate (r) works on the input storage management, as described by the following equation:

$$r_n = \sigma([W_r x_n] + [U_r h_{n-1}]) \quad (3.10)$$

where the W_r and U_r are the weights that manage the synapses from the input and last state, respectively. The σ function allows only values of 0 and 1, and, therefore it serves as a switch between the input or the last state and is used to compute the candidate for the next state (\tilde{h}):

$$\tilde{h}_n = \tanh ([W_h x_n] + [U_h (r_n \cdot h_{n-1})]) \quad (3.11)$$

where W_h and U_h are the weights that manage the information from the input and information of the last state that the reset gate allows, respectively. Consequently, the reset gate controls the frequency of the storage containers, as it controls directly what synapses should pass at input. Therefore, when this gate is closed, the next state candidate will only be calculated based on the last state [59, 60].

The output gate (z) is responsible for controlling the next state of the network, as the following equation suggests:

$$z_n = \sigma ([W_z x_n] + [U_z h_{n-1}]) \quad (3.12)$$

where W_z and U_z are the weights that manage whether the information is passed on from the current input or the last state. Even though it is similar to the r gate, the difference lies in how the next state (h_n) is computed:

$$h_n = [z_n \cdot h_{n-1} + (1 - z_n) \tilde{h}_n] \quad (3.13)$$

Hence, the state computation chooses between the input synapses or the computed candidate based on the update gate output. If the network is updated, a candidate is generated based on new information, while not updating will result in the maintenance of the previous cell state. This makes this gate responsible for managing the long-term memory, as it selects between the pre-selection that the r gate made and the new information that should participate in the new state [59, 60].

LSTM and GRU are used in a vast area of applications, namely protein secondary structure prediction [61], acoustic modeling of speech [62] and music [15, 63], speech synthesis [64] and recognition [65], audio analysis [66], and handwriting recognition [67, 68]. There is also a branch of study that uses GRU models in the area of language comprehension for text translation [69], text generation [56] and image and video description [70–72].

3.6 Architecture Paradigms

Different configurations of the mentioned architectures produce interesting results based on the question to be solved. Paradigms for the encoding of the input in a latent space by using autoencoders, generation of original content by using Generative Adversarial Networks (GAN), and decision mechanisms that depend on the environment by the use of Reinforcement Learning (RL) are examples of configurations that help solving problems in the real world.

3.6.1 Autoencoders

This deep learning architecture is a non-supervised learning NN that reproduces the input signal by encoding the input and generating an output from the encoded part. Fig. 3.8

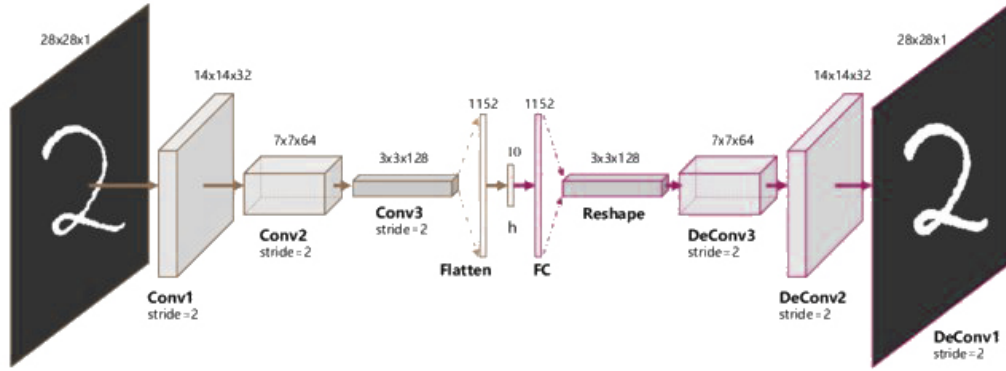


Figure 3.8: Example of an autoencoder for reproducing images of numbers. From [73]

shows an example from [73], where three convoluted layers produce a latent space vector that contains the narrowed encoded information needed for the generator to decode and reproduce the content. Some applications involve de-noising images [74], compressing information [75], and even learning features [73]. These features can be used for other clustering and classification algorithms [73].

Other variations of this algorithm include the Variational Autoencoder (VAE) [76, 77] and Weighted VAE which learns the statistical distribution of the input. The output, which is also a replica of the input, is not generated directly by the raw data, but from the learned parameters. As the encoder learns values such as the mean and standard deviation, the generator understands how they impact the output.

3.6.2 Reinforcement Learning

Reinforcement learning is the training of ML models to take actions based on the feedback given by a complex environment (Fig. 3.9). Often these algorithms are compared to a game, where the agent is penalized or rewarded according to a specific objective. This reward system gives encouragement or inhibition for performing an action in a given state and the end-goal is to obtain the highest reward possible.

Even if the rules for obtaining points are given by the user, it is the model that learns how to perform better to maximize the results. This may lead to sophisticated tactics that a human being couldn't conceive. An example of the achievements made by DNN RL was when AlphaGo defeated the European Go champion in 2015. It was a renowned mark in AI evolution as even though the rules of the game are simple, theoretically there are 10^{700} possible ways to play the game, therefore playing aided by brute-force computing is not feasible [78].

3.6.3 Generative Adversary Networks

Generative adversarial networks is a generative model that is pitted against an adversary. This adversary is a discriminative model that learns if a sample is from a model output

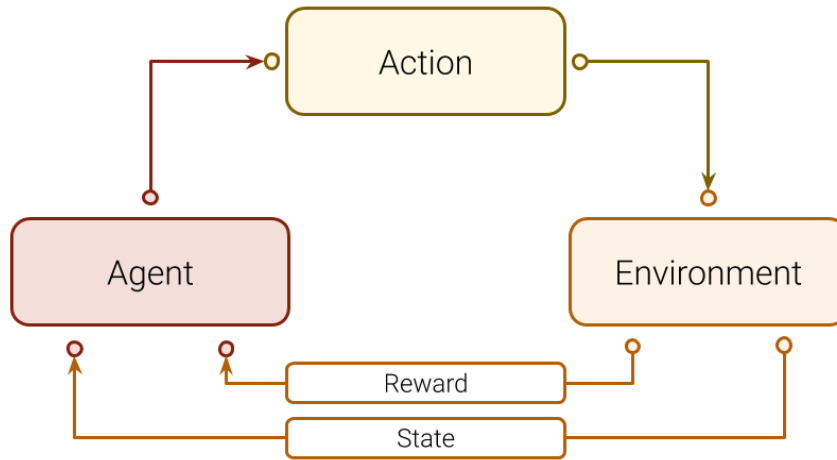


Figure 3.9: Diagram for the Reinforcement Learning paradigm. The agent produces actions on the environment, which reacts causing a change of state and giving a reward to the agent. Based on what the agent sensed it performs a new action.

or from a data distribution. The generative model can be considered as a counterfeit that tries to produce fake data in order to trick the discriminator to believe it is real. Consequently, while the rivals compete, the generator generates counterfeits that can not be distinguished from the original objects [52].

Applications for generating data with GAN have been used in the creation of art, as primary data for training other algorithms or to impose superresolution. Some of the most impressive contributions by these algorithms were made by Karras, Laine, and Aila (2018) [79] with Style GAN where faces of real people are merged, resulting in the creation of high-realistic, non-existent faces. Another example of the application of GAN is "Deep Fake", which manipulates images and videos. The very realistic reproduction of the created media is the result of merging, combination, and replacing of images and videos from different sources. This application has suffered serious criticism due to the possibilities of serious forgery that could damage societies and politics [80].

3.7 Optimization

The goal of training is to approximate a function to return the most accurate prediction given an input. The optimization algorithm solves the problem where the objective function is minimized or maximized. In the case of NN, commonly this objective is the function that gives the error of the prediction, called the loss function, and is dependant on the model's internal parameter values. These parameters comprise the weights and bias of all the network and how they interrelated with each other through the graphs that connect each neuron. When updating these values, the calculated output will change for the same

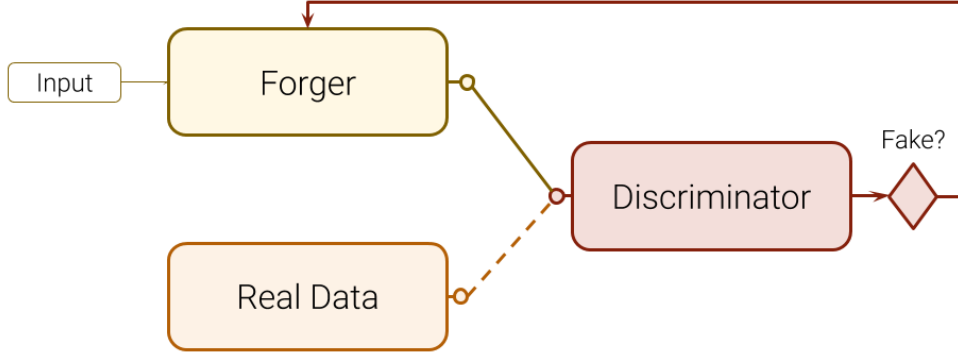


Figure 3.10: Diagram of a Generative Adversary Network. As the forger produces fake data, the discriminator will decide the truthfulness of what is given as input. This information is given to the forger that produces more convincing data, while the discriminator becomes better at recognizing fakes.

input. Therefore, the optimization is the process that changes these parameters in a way that in each iteration, the networks are capable of simulating the objective function.

The optimization algorithms play a major role in how the training is processed in terms of performance, speed, and computational resources [81]. Following this reasoning, the desired optimum parameter value $\hat{\theta}$ is when the loss, which depends on the parameters, is minimum :

$$\hat{\theta}_i = \arg \min_{\theta} \left(\frac{\partial L(\theta)}{\partial \theta_i} \right) \quad (3.14)$$

the gradient will be represented in the next equations as $\nabla L(\theta)$:

$$\nabla L(\theta_i) = \frac{\partial L(\theta_i)}{\partial \theta_i} \quad (3.15)$$

The first order algorithms minimize the Loss function using only the gradient values with respect to the parameters, such as the Stochastic Gradient Descent (STG). They use the first derivative in order to calculate the gradient. The second-order algorithms use the Hessian matrix, i.e. second-order partial derivatives matrix, that allows detecting the curvature at that point. Since the second-order algorithms are much more costly in terms of computation requirements, we will only focus on the first order ones[81].

The most common loss functions for DNN are the mean squared error, mainly used for regression, and cross-entropy, mainly for classification. These loss functions are, respectively, given as follows:

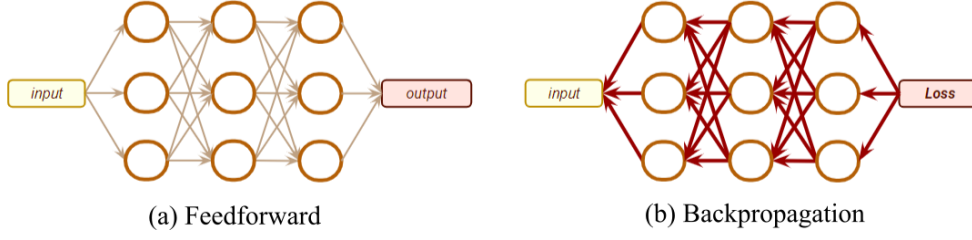


Figure 3.11: Flow of calculus in each optimization step.

$$L = \frac{1}{N} \sum_n^N (y_n - \hat{y}_n)^2 \quad (3.16)$$

and

$$L = -\frac{1}{N} \sum_n^N [y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \quad (3.17)$$

3.7.1 Backpropagation

In order to optimize a complex function, iterations of two main steps are made: error calculation and weight update. The first involves the forward propagation of the input through ANN to generate the output (Fig. 3.11). This is followed by the backward propagation of the ANN. The difference between the targeted and actual output values is used to calculate the gradient of the loss function in respect with the weight. In the second phase, the weights are updated by removing a percentage of that gradient. This process is repeated until the performance of the network is satisfactory [82].

Backpropagation is one of the most used optimization algorithms for NN but, as discussed before, it is prone to vanishing and exploding gradients.

3.7.2 Vanishing and Exploding Gradients

When a network has h hidden layers the information that is propagated through the network undergoes affine transformations followed by the activation function. When the computation of a derivative has to be made, through the backpropagation method, based on the previous h layers a chain-rule is formed. Therefore two situations may present themselves, the first is when these derivatives have an added effect, leading the gradient to "cliffs" of the loss function, that make it explode into very large numbers. The second effect is when the derivatives produce an attenuation, making the gradients into zero [83].

For example, sigmoid and hyperbolic tangent functions are prone to vanishing gradients when used in DNN. As seen in Fig. 3.12 when a sigmoid function is submitted to successive derivatives it tends to attenuate the effect of the gradient while training.

Several rules can be used to control these issues, such as normalization of the loss with L1 (eq. 3.18) and L2 (eq. 3.19) and clipping the gradient to a threshold value.

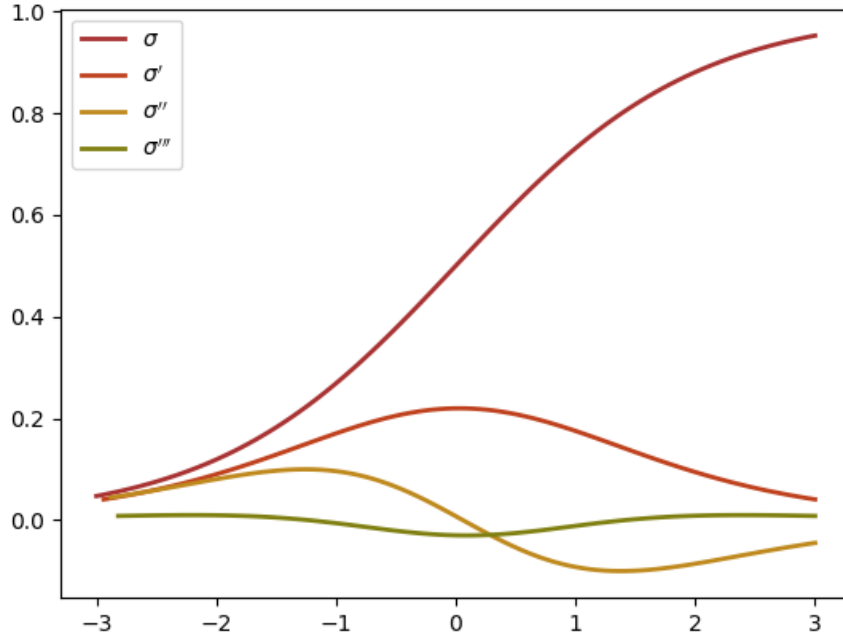


Figure 3.12: Sigmoid function successive derivatives graphical representation.

$$L1 = \sum_{i=1}^N \|y_i - f(x_i)\| \quad (3.18)$$

$$L2 = \sum_{i=1}^N (y_i - f(x_i))^2 \quad (3.19)$$

3.7.3 Network Initialization

The weights must be first initialized in order for the optimization step to be successful. One of the most simple ways to initialize is to just give normal distribution weights with an average of 0 and a standard deviation of 1. The downside to this approach is that in some cases it is counter-productive and makes it harder for the optimization algorithm to converge. One of "tricks of trade" is to keep the standard deviation of the outputs of activations around 1.

Depending on the activation functions, two different algorithms will be presented as being efficient. The first is the Xavier Initialization. This is another common method for the hyperbolic tangent activation and even sigmoid functions. This method implies giving a variance of $\frac{1}{\sqrt{h}}$ to the Gaussian distribution, where h is the number of neuron numbers. This is a method that helps the outputs to be stable [84].

As for the ReLU initialization, even though the Xavier Initialization, He et. al. (2015) [85] described that a network would converge faster if the following strategy applied :

- initialize weights with a standard normal distribution;
- multiply weights by $\sqrt{\frac{2}{f}}$, where f is the "fan-in", i.e. the number of incoming connections from the previous layer output;
- the bias tensors are initialized to zero.

Therefore as the networks become deeper, some initialization strategies should be applied in order for the algorithm to converge.

3.7.4 Gradient Descent Algorithms

Large-scale computational complexity relies on optimization in non-trivial ways. Even though a simple Stochastic Gradient Descent (SGD) is a method that shows good performance for large datasets [86], other methods exist in order to help to converge faster when the first method does not work properly. Momentum [87], Nesterov accelerated gradient [88], Adagrad [89], Adadelta [90], Adam [77] and RMSProp [91] are some examples of algorithms that are provided for parameter optimization. Since the algorithms that are used throughout this thesis are SGD, Adam and RMSProp, these are the ones that will be explained in detail.

3.7.4.1 Gradient Descent

Learning with gradient descent may be represented in the following simplified equation:

$$\theta_e = \theta_{e-1} - \eta \nabla L(\theta) \quad (3.20)$$

where θ are the parameters to optimize, e the epoch, i.e. the iteration, $\nabla L(\theta)$ the gradient of the loss function in respect to the parameters and η is the learning rate. This last variable plays a major role while training a network, and several algorithms change its value in order to reach better results. If the learning rate is too high then the function will not converge, because it takes bigger steps when changing value, resulting in massive changes to the weights and the loss can potentially go higher. When its too low, the convergence may take too long to be considered and sometimes the descent may be stuck in a local minimum, whereas higher learning rate values may help the convergence to surpass a local minimum and continue the descend through the loss function.

3.7.4.2 Adam

The Adaptive Moment Estimation (Adam) [92] is an algorithm for first-order gradient-based optimization of stochastic objective functions that work by adapting the learning rate for each parameter by using the momentum. There are two variables, called momentums, that need to be calculated, m_e that can be seen as the mean and v_e that can be seen as uncentered variance and can be calculated as:

$$m_e = \beta_1 m_{e-1} + (1 - \beta_1) \nabla L(\theta)_e \quad (3.21)$$

and

$$v_e = \beta_2 v_{e-1} + (1 - \beta_2) (\nabla L(\theta)_e)^2 \quad (3.22)$$

the β_1 and β_2 variables are hyper-parameters, which the deep learning community usually adopts the values of 0.9 and 0.999, respectively. The second step of this algorithm is calculating the estimated \hat{m}_e and \hat{v}_e :

$$\hat{m}_e = \frac{m_e}{1 - \beta_1} \quad (3.23)$$

and

$$\hat{v}_e = \frac{v_e}{1 - \beta_2} \quad (3.24)$$

and the last step is to update the weights using these variables:

$$\theta_e = \theta_{e-1} - \eta \frac{\hat{m}_e}{\sqrt{\hat{v}_e} + \epsilon} \quad (3.25)$$

the unmentioned variable ϵ is a very small number that makes it harder for the denominator of the fraction to become zero.

3.7.4.3 RMSProp

RMSprop is an unpublished optimization proposed by Geoff Hinton in a lecture of the online course “Neural Networks for Machine Learning” [91]. This encompasses an optimization that is based on a decay factor γ , so for each epoch e while optimizing, it follows the succeeding rules:

$$\theta_e = \theta_{e-1} - \frac{\eta}{\sqrt{E[(\nabla L(\theta))^2]_t + \epsilon}} (\nabla L(\theta))_t \quad (3.26)$$

where $E[(\nabla L(\theta))^2]_t$ is the estimation of the loss function gradient for the epoch e and is calculated as:

$$E[(\nabla L(\theta))^2]_t = \gamma \cdot E[(\nabla L(\theta))^2]_{t-1} + (1 - \gamma) \cdot (\nabla L(\theta))_t^2 \quad (3.27)$$

the common values for γ is usually between 0.9 and 0.95 [93].

Throughout the practical applications of this thesis, as an adaptation of the RMSProp is used, where the loss function is adjusted depending on the evolution of the loss during training. More detailed information will be held in Chapter 8.

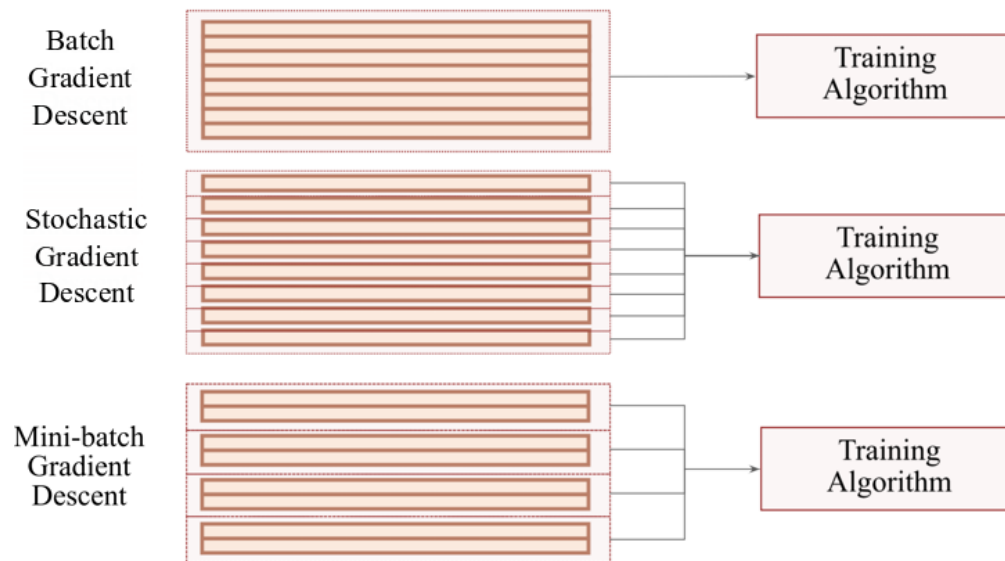


Figure 3.13: Diagram of the three batch training approaches. Each orange rectangle is an atomic record of data. In the batch gradient descent all the data is fed to the training algorithm in one batch. The stochastic gradient descent each record is fed in sequence into the training algorithm. At last, the mini-batch gradient descent divides the data into a number of batches with a fixed size and each batch is fed sequentially into the algorithm.

3.7.5 Batch Training Approaches

There are three main approaches to train data in batches: Batch, stochastic and mini-batch gradient descents. Figure 3.13 depicts a diagram with these methods, which is based on how data is organized when fed into the network while training. All these methods will be described briefly.

3.7.5.1 Batch Gradient Descent

The first is the batch gradient descent, also called "vanilla", and deals with all data as a whole for each epoch. This approach is very slow in each epoch and the descent will most probably lead to local minima and, consequently, not leading to the minimum loss value possible. Each epoch also tends to be redundant, as it is dealing always with the same data. Another limitation is that the computation requires a lot of working memory in order to manage high quantities of data at the same time, resulting in it being impracticable most of the times.

3.7.5.2 Stochastic Gradient Descent

The second, the stochastic may be seen as the training of each record at a time. Even though it permits to jump over local minima reaching lower loss values. When using this algorithm, before each epoch starts, the data is shuffled so that the same order does not repeat, removing the redundancy. This algorithm, unfortunately, can lead to high variance

of the output due to low abstracted representation of the data, as the weights can record the memory for each record.

3.7.5.3 Mini-batch Gradient Descent

The third batch method is the mini-batch, opposed to the stochastic gradient descent, each training epoch is dealt with chunks of data (batches), and, opposed to the batch gradient descent, there is number of equally sized batches. After each epoch, all data is again re-organized randomly into new mini-batches of the same size. Since the combination of each batch is made differently, the network gains a higher abstraction than when trained with the stochastic gradient descent. Another advantage of this method is that the DNN libraries exploits the maths making its computation more efficient when using GPU. Therefore, using this approach makes training faster when dealing with high amounts of data.

3.7.6 Cross-validation

When dealing with ML algorithms cross-validation is important in validating the generalization performance of the algorithm. The separation of data in training, testing and, in some cases, validation sets is crucial for assessing how the results have statistical importance when validating an independent dataset. The training set is used for training the model, the testing set is used to test the trained model with data that the model did not see before, and validation is also not seen by the model while training, but it is used to validate how the model is evolving in each epoch. Overfitting presents itself as a problem when cross-validation is not performed, as the models adjust themselves to the training data, deviating from the intended generalization.

The various technical differences between how the cross-validation is made are in what percentage is allocated to each set. The leave-one-out method takes all the examples, except for one, to be trained while one record is used for testing purposes. Recurrently this is done for the whole dataset, wherein each iteration the test record is different. The results are based on the statistical average of the prediction analysis outputs. In most DNN approaches this is impractical because when dealing with large datasets, the computational burden is too high for it to be considered.

The k -fold is a method that separates the data into k different batches. Just like before, each fold is processed as a testing group, but the other batches are used for training. The folds are only randomized at the start and this is repeated until all the folds are covered. This can be used for DNN problems as long as the model does not take too long to train, due to too many records and/or the model complexity.

The most common approach to randomize the records of the dataset at the beginning of the dataset and give a percentage for each segment. The performance of the algorithm should be evaluated while training, as the problem of overfitting is common in DNN. One way to control this problem is to plot two curves, one for the training set and the other in the validation set, as displayed in Fig. 3.14. This graphic depicts a typical behaviour of

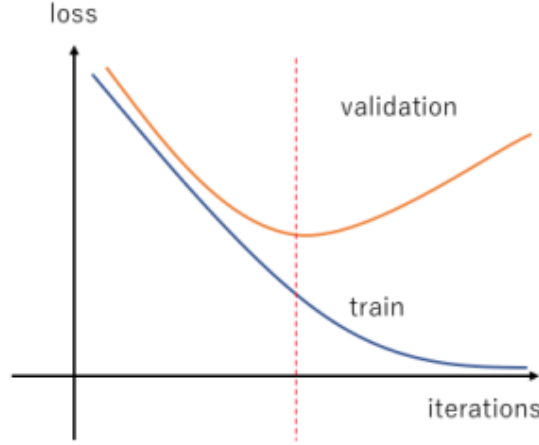


Figure 3.14: Standard curve while training. Both curves represent two sets, the training and validation sets, while the model is adjusting its parameters throughout the epochs.

the loss curve throughout the number of iterations. Both curves drop significantly in the start while decreasing the slope until reaching a limit value. In the displayed example, in contrast to the training set, the validation set starts to rise, this means that the weights no longer are optimized to the problem, but to the training data. Therefore, the training should be stopped when this inflection of the validation curve occurs.

3.8 Models Evaluation

For classification modalities, the models' evaluation will be given by the following equations:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3.28)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (3.29)$$

$$Specificity = \frac{TN}{TN + FP} \quad (3.30)$$

where TP, TN, FP, and FN respectively stand for True Positives (when the sample is classified as the evaluated class correctly), True Negatives (when the sample is classified as not being part of the evaluated class correctly), False Positives (when the sample is classified as the evaluated class incorrectly) and False Negatives (when the sample is classified as not being part of the evaluated class incorrectly).

The False Negative Rate (FNR) and False Positive Rate (FPR) are computed by:

$$FNR = \frac{FN}{TP + FN} \quad (3.31)$$

and

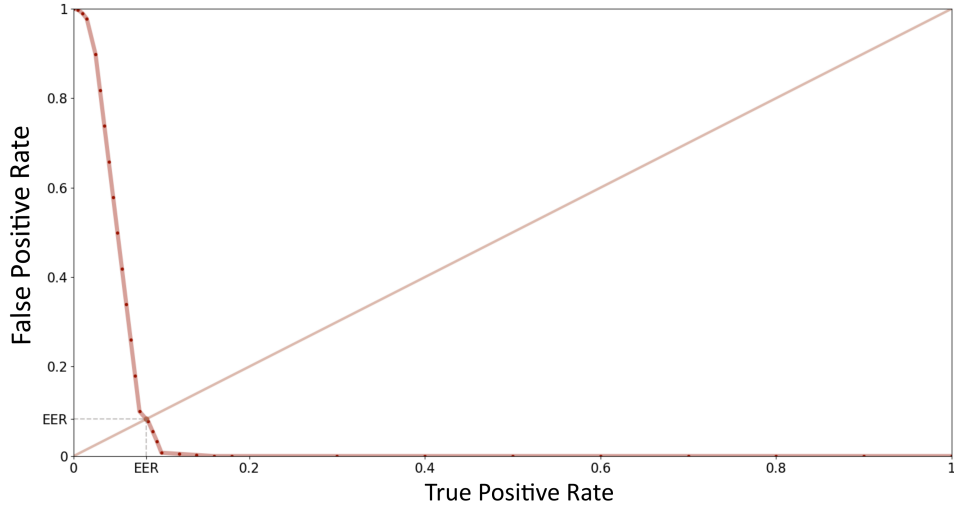


Figure 3.15: Example of a ROC curve and the point in which the EER is calculated.

$$FPR = \frac{TN}{TN + FP} \quad (3.32)$$

these measurements check the rate in which the samples are wrongly classified against all the true values, i.e. the FN concerning all the "being of the evaluated class" for the FNR and the TN about all the "not being of the evaluated class" for the FPR.

When a binary classifier for a specific class is subjected to a classification threshold, the FNR vs FPR values may change. The graphical representation of those changes when submitted to an array of thresholds is named Receiver Operating Characteristic (ROC) and an example may be observed in Fig. 3.15. The Equal Error rate (EER) measurement is when the FNR = FPR. For several ROC the mean and standard deviation may be calculated and used as a validation for several binary classifiers.

3.9 The parallelism between real and artificial neural networks

The Artificial neural networks (ANN) inspiration in the biological nervous system goes beyond the neuron. From the simpler version of the mechanism of the synapse to the more complex memory, language and image processing of the central nervous system, the ANN has attempted to produce mathematical analogies that could model and mimic these systems. During its development it has been able to reach peaks in performance in both research and industry. This section will refer to some aspects and details of the architectural concepts that analogs nature behaviour.

3.9.1 Postsynaptic Synapses and ANN action potential

Before the synapse, the electrical signal which stimulates each dendrite increases the electric potential inside the cell, until reaching a threshold value. There is also the possibility for inhibitory and facilitator events (refer to section 2.6.1.3). In the reality of ANN each perception has an input, that can be stimulated, or not, depending on the input. Each input is then multiplied by a weight attributed to its own vertex, between two graphs. These weights are the facilitators or inhibitor channels, as they coordinate and regulate the signals that pass through the network. Another interesting fact is that the bias, which is summed after the internal product of the input with the weights, can also be seen as the action potential threshold that exist in the real neuron. With the increase of the bias value the higher the input values or the weights that the neuron needs to be triggered.

3.9.2 Brain Cortex and pathways

The brain cortex is layered into six different layers (refer to section 2.6.4), as the brain also has areas that process the information and others that make sense out the processed information. The pathways connect these areas for a specific purpose such as processing visual and memory information. These pathways can be related to the feature extraction mechanisms that then are passed to the classification part of the network, and the latent information is passed to the last layers, where the decision is made.

The cerebral cortex has pathways that interconnect different areas (corticortical connections), such as the connection between the association cortices, primary and secondary sensory, and motor cortices. Therefore in order to improve current neural networks this is a mechanism that could be mimicked in order to different networks to share information in a standard way. The neural networks architectures could be organized to improve the complexity of the classification layers and make more informed decisions of a classification paradigm. If the fusion layers provide intersections between two networks for different types of data, some interconnections may be used to augment the learned knowledge and infer about the process that is at hand. For example, the connection between networks with the knowledge of speech sounds and a network with knowledge of videos of faces while speaking, they can be joined together with fusion layers and give a more informed about the output subtitles.

3.9.3 Memory Processing and Recall

Section 2.6.6 refers to this topic. The known memory processing is through the facilities due to the passage of a larger number of times to a specific goal, the ANN memory units are the weights, that are built over a number of times an input is given. Therefore in order to increase the memory of the network, the number of hidden units should be increased, so that the number of weights increase. Also in this regard, the facilitated pathways may

be also be compared to the ANN attention mechanisms, where the features are highly regarded in detriment of others.

These pathways can also be triggered in order to generate synapses, even in the absence of a sensory direct stimulus, generating an experience recall. This is analogous to the generative architectures, as the network learns, it can replicate it given the right input. The exposition to the same similar experiences, the more concrete generations it is possible can make.

In the learning processes, the brain uses the the old memories to help to create new pathways, when new information and context is given. As it performs a conjoint operation of assimilating new memories with the previous knowledge it processes. Transfer learning is also made when the exposition of new information is made in pretrained networks with preexistent knowledge. Nowadays a vast range of applications, namely computer vision, uses pre-learned filters of images and videos, and use them for more specific purposes or learn others by fine-tuning their filters.

Sequential data is also highly depend on knowledge, such as language. The previous data is important to understand the whole context and the next data. The RNN also has this particularity, as it relies on the previous information to make sense of the full context. Several studies connect networks in the context of text-processing and speech detection to encode the relevant pieces of speech and combine them to text. Translation mechanisms also rely on this memory, as the RNN encodes the sequence in one language, the latent vector with the attention layers makes sense of the core of the transmitted information into a new language. Therefore, the pieces of information must be sequential in the way it is learned, otherwise the context of the whole is lost, such as the perception and even formation of speech in the brain.

Other mechanisms such as procedure knowledge (learned habits) and declarative knowledge (familiar experiences) are also important factors to the storage of information. The use of the rewarded or punished consequence provided by the amygdala creates the behaviors that the human will learn and remember. In terms of learning through experiences it is possible to see that in RL there can also be two networks, one for the policy, i.e. the procedure one may take given experience, and other for the interpretation of the information given through familiar experiences. Both these networks are rewarded or punished by the environment through a score based on the behavior the network chooses to apply.

3.9.4 Visual Sensing and Processing

Section 2.6.7 describes that before reaching the cortex, the information is preprocessed, such as the need to stabilize and normalize the data before reaching the network and the brain cortex also has the configuration in layers where the input information is distilled the deeper it goes.

The light intensity such as brightness is also a feature that is extracted through the edge filters that are learned in image processing. These are responsible for detecting where an

object end. When separating the image into the channels of RGB, they provide information in a related analogy as the cones provide that information. Therefore the filters are learned in these different layers of the image such as angles, lines or a specific length, angular shapes like the visual cortex. Deep Learning (DL) architectures are layered, and each layer extracts less detailed information as it progresses during the abstract feature processing module. This mechanism is parallel to the layers of the primary visual cortex, in which its processing pathways comprise specialist cells that are activated once that specific color or form is detected.

The vector of synapses progresses to the secondary visual cortex, where that information is interpreted, an analogy to the latent vector which is carried to the next sequential modules of the network. These may refer to the classification or generative layers, where the patterns are detected and output made based on the combination of the previously encoded data.

In order to pass on to the next stage of DL some mechanisms could be looked upon and improved closely to how our own biological structure works. As we understand better how the central nervous system regulates and learns, we will be able to build higher cognitive structures and proceed to the next generation of AI.

State-of-the-art

This chapter has two sections, the first is regarded to state-of-the-art developments made in Deep Neural Networks (DNN) technologies used in biosignal applications, while the second refers to the state-of-the-art referring to the application scenarios described in Chapter 8. The first section includes a summary and discussion of the most relevant works that were found in Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN) and other DNN fields. The second section is organized in three application scenarios: biosignal synthesis, biometric solutions and abnormality detection of Electrocardiogram (ECG) signals. In this section, other Machine Learning (ML) algorithms and results for each particular scenarios are presented.

4.1 Application of DNN

Research regarding the use of DNN for images, videos and sequences is very active since 2013, whereas the biosignal community started to publish more intensely from 2015 onwards. The relevant examples were Rodrigues and Couto (2012) [94], Martinez, Bengio, and Yannakakis (2013) [95] and Zhu et al. (2014) [96]. The authors Rodrigues and Couto (2012) [94] proposed a deep neural network approach for ECG denoising by using a feed forward method with three hidden layers. As for the second work uses an CNN autoencoder to extract features from Blood Volume Pressure (BVP) and Electrodermal Activity (EDA) and a CNN classifier in order to establish a correlation between four emotional states (relaxation, anxiety, excitement and fun) and physiological changes for both these signals. The human-made features were outperformed by the machine-made features [95]. This application was put into action using records with electrode motion artifact noise added. While achieving a good performance with high level noise, a minimal information loss was detected in low level noise. This research showed promising advantages for applying Holter

records, despite not being applicable in real time [94]. The last, Zhu et al. (2014) [96] used a two layered CNN for detection of drowsiness from Electrooculogram (EOG) signals.

4.1.1 Application of CNN

Most of the papers related to CNN from 2015 to 2019 are application examples of the CNN. Some of the early works are direct application of previous 2D CNN to images, by changing 1D data into 2D or by extracting a spectrogram that gives information on frequency in relation to time. More recently, some approaches that uses these networks to extract features directly from the raw signals started to appear and will be described in the following sections.

4.1.1.1 CNN applications based on feature engineering

A scenario where the raw signal is transformed into a color coding scheme was found in the work of Geng et al. (2016) [97]. In this work, the Electromyogram (EMG) from multiple sensors were converted into physical actions, which were recognized by a prosthetic hand. Each sensor recorded data of several muscles from amputees represented in a matrix where each row represented an EMG channel while each column a timestamp sample which was converted to an image where each value was substituted by a color from a colormap. The prosthetic hand understood 8, 24 and 52 different gestures, achieving 99%, 95% and 97% of accuracy, respectively, by majority voting. Unfortunately the architecture is not explained in detail [97].

Several applications are found to make analysis in the frequency domain by the use of spectrograms in the area of EMG [98], Electroencephalogram (EEG)[99], ECG[100–102] and Phonocardiogram (PCG) [103–105]. In the work conducted by Cote-Allard et al. (2017) [98], the EMG are used to detect seven hand gestures with a pre-trained model achieving 98 % accuracy. Xia, Hu, and Peng (2018) [106] also uses CNN extracted features from images but uses RNN to classify sequences in the field of EMG for detecting movements [106]. Sengur et al. (2017) [107] uses Wavelet transform spectrograms to detect neuromuscular amyotrophic lateral sclerosis with 97% accuracy.

In the domain of EEG, Tsinalis et al. (2016) [99] had an overall accuracy of 74% to detect changes in different sleep stages by using a simple CNN-based architecture. One interesting aspect of this study is that, after observing the learned filters, these were similar to the patterns that follow the American Academy of Sleep Medicine. Therefore, the experts could relate the mechanisms that the network learned with the theory.

Xia et al. (2018) [100] applied a regular CNN on a fusion between a spectrogram obtained with a FFT and the wave transformation through stationary wavelet transform and used a regular CNN for the detection of atrial fibrillation in ECG with an accuracy of 99%. A very interesting study, conducted by da Silva Luz et al. (2018) [102], used an architecture that fuses a 1D CNN, fed with raw ECG heartbeats, with another 2D CNN

fed with the corresponding spectrograms, reaching state-of-the-art performance on off-the-person ECG biometrics. Also in the realm of biometry, Zhang, Zhou, and Zeng (2017) [108] was able to use wavelet transform for the identification of people using wearable technologies.

In one of the PCG studies, Dominguez-Morales et al. (2017) [103] converted the raw signal into an address-event representation and later applied 2D sonogram. This images were input into a modified version of the AlexNet [109] reaching 97% accuracy in detecting pathologies [103]. Other studies, such as Nilanon et al. (2016) [104] and Pan et al. (2017) [105] had accuracies above 90%.

The approaches presented until now solely used spectrograms or converted images from data. These, being possible to use for the mentioned purposes, may not capture the essence of biosignals. Therefore, information that could be fundamental to the detection of relevant characteristics not perceptible by the human expert may be lost. One study that also includes information directly from the raw signal was da Silva Luz et al. (2018) [102], which enriches the input for the network, helping the decision process.

Additional features, beside spectrograms, can be extracted and the CNN is used as a classifier. Several examples of this paradigm for enhancement, detection, diagnostics and prediction. For example, in the ECG domain, features such as Heart Rate Variability (HRV), extracted from the RR interval, can be used for several purposes, such as congestive heart failure detection [110]. Another example can be seen in the area of PCG, where Maknickas and Maknickas (2017) [111] used several features, such as Mel-frequency spectral coefficients, to detect normal/abnormal signals, being able to reach an accuracy of 84.15%, by means of a 2 layered CNN.

4.1.1.2 Application of CNN to raw data

In contrast with the works presented in the previous section, several authors have been using CNN directly on data with a low pre-processing level in order to extract machine learned features. The aforementioned works of Martinez, Bengio, and Yannakakis (2013) and Zhu et al. (2014) [96] are the first examples of this methodology which used BVP and EDA to establish a correlation between them and four emotional states.

Additionally, more recent works are found to apply this approach to other biosignals, but mostly to ECG signals. Al Rahhal et al. (2016) [112] used an autoencoder to learn features and applied a fully connected layer with a softmax activation to classify the learned features. They also studied how the number of hidden layers changed the accuracy reaching results above 90% of accuracy for different databases.

Yao, Zhu, and Chen (2017) [113] developed an architecture where parallel CNN models analysed different scales of the signal and concatenated the learned characteristics for the detection of atrial fibrillation and achieved 98% accuracy on the MIT-BIH database [114]. In the same database, the discovery of abnormal ECG events Zhou, Jin, and Dong (2017) [115] suggested a multi-lead CNN model, together with an inference rule to improve

performance, reaching 99% of accuracy. Liu et al. (2018) [116] also proposed a method that use several leads and extract the information from the QRS complexes to improve real-time myocardial infarction detection.

More recently, Acharya et al. (2018) [117] used CNN composed of 11 layers, analysing segments of ECG with either 2 or 5 seconds, instead of the common technique of using only one ECG beat. This architecture classified between normal, atrial fibrillation, atrial flutter and ventricular fibrillation. However the lack of examples for signals with ventricular fibrillation faulted more than a third of these cases to be wrongly classified. Good results in terms of accuracy, sensitivity and specificity were achieved. Nonetheless, there was a need for pre-processing the signals in order to remove noise and the high amount of data required, consequently the amount of time for training the network was considerably increased [117].

Savalia and Emamian (2018) [118] proposed a deep learning algorithm that consisted in a fully-connected network, containing 4 hidden layers, and a CNN with 4 layers. It was used to predict between a normal or abnormal sinus rhythm, while the CNN performed preprocessing, feature extraction and prediction between 9 different classes. The test results showed high accuracy of 88.7% for the fully-connected network and 83.5% for CNN, even though some confusion between atrioventricular block and ventricular bigeminy was detected, due to similar features [118].

State-of-the-art results in detecting arrhythmia are currently in the record of Hannun et al. (2019) [119], as they developed a 34 layer DNN that could detect between various arrhythmias, normal signal rhythm (NSR) and noise corrupted segments, constituting a total of 12 classes. The raw data from single-lead ECG signals is fed into the network and had results that were reported being higher than the average cardiologists diagnostics producing an area under the curve of 0.97 for both confusion matrices (DNN and specialist). This means that most misclassifications made by the classifier were also made by manual reviews by specialized physicians that can be justified by lack of context from other historical events, limited signal duration and being just an analysis from a single lead [119].

Most of the research using ECG and CNN is based on the detection of symptomatic events from ECG signals. CNN architectures are also applied for denoising such as Ansari, Gryak, and Najarian (2018) [120]. In this study, after learning the morphology of the signal in the training phase, the proposed algorithm segmented the signal and predicted if they would be usable for HRV calculation, achieving lower error rates when applied in this task. This CNN algorithm was used for ECG noise detection, but only concerned with QRS peak detection. The segments that were classified usable for further processing had a better probability of containing certain levels of noise incapacitating other types of classification methods [120].

Other biosignals such as Respiration Signal (RESP) were used for emotion recognition [121], PCG for heart anomaly detection [122], EMG for classification of hand movements from the arm contractions [123], speech reproduction from silent speech [124], and accelerography for human activity recognition [125, 126] were extracted using raw signal

signatures.

Most of the CNN architectures seen, except for Hannun et al. (2019), used simple networks of a few layers that extract features and/or do the classification process. The signal is usually segmented and filters that remove noise are employed in most of them. In the case of more than one sensor reading, normally several networks are trained, each for each channel and then a fusion layer is added to the end, right before the classification layers. Some of them employ modified versions of the LeCun [49] or AlexNet [109] networks used extensively in image and object recognition.

4.1.2 Application of RNN

Examples applications of RNN have been used in the context of apnea detection from extracted RR signal [127] and powerline elimination using RNN applied to ECG [128] in order to enhance the signal.

In the work of Chen and Jin (2015) [129] a feature fusion method is applied, where features from audio, image, ECG and EDA signals are extracted in order to detect arousal and valence of emotion by means of a three layered Long Short-Term Memory (LSTM). One work from Thill et al. (2019) [130] uses the interesting approach of stacking LSTM in order to detect anomalies in the ECG. Firstly, the network learns how the signal behaves. Then, by submitting a new segment of the signal, a prediction error that can be fit with a Gaussian model is given. The final classification is based in the minimum Mahalanobis distance.

Also, in the case of arrhythmia detection, in this case for premature ventricular contraction, Zhou, Jin, and Dong (2017) [115] used LSTM together with an inference rule. Two databases were used achieving accuracy above 98% with a dataset with over 140 000 ECG recordings. Several studies used the extraction of features from CNN structures. For example S. Andersen, Peimankar, and Puthusserypady (2018) [131] classifies atrial fibrillation episodes in segments of Holter ECG recordings of 24 hours. The CNN extracts features from the RR interval time-series and is fed into the LSTM network. The combination of these networks allowed to achieve a higher classification accuracy in high efficient time scale, as predictions of 24 hours of ECG were made in one single second. Despite being a promising deep learning approach, the noise present in some segments affected the classification [131].

Thomae and Dominik (2016) [132] also used the combination of CNN and RNN architectures for PCG signals, while adding an attention mechanism, followed by a fully connected network for the final classification. In addition, in the domain of PCG, study has been conducted by Schölzel and Dominik (2016) [133], where they used this biosignal and a RNN to synthesize ECG. They experimented with seven networks, in which five of them were inspired by sequence-to-sequence network [134, 135] and had the following structure: the first was a simple feed-forward network; the second a bidirectional LSTM (one for each way); the third was a four layered (for each way) bidirectional LSTM; and one simple

LSTM with using an evolutionary algorithm for training (instead of backpropagation), and at last; a five layered LSTM with low number of hidden neurons. Even though they produced good results, they acknowledged the difficulty in reproducing the ECG signal, namely the placement of the wave. In the field of EEG, Dong et al. (2018) [136] uses an LSTM approach for detecting sleep stage progression using several frequency features, while detecting which was the best electrode placement for that analysis.

These networks are used for learning the mechanisms behind sequences of features, time-windows or samples. Due to the difficulties behind the learning process, feature extraction mechanisms are implemented. The use of autoencoders and CNN to extract these features are a trend that may help surpassing these challenges.

4.1.3 Other Networks

Some simpler architectures were also subjected to the classification of symptomatic episodes, applying directly DNN classification methods using human-extracted features. For example, an architecture composed by 6 layers of full connected network was created by Jun et al. (2016) [137] for detection of premature ventricular contraction. Only using common features used in the field of ECG processing such as amplitude of the R peak, distances between the characteristic waves maxima and area of the QRS complex, they achieved 99% accuracy and a sensitivity of 96%.

The deep belief networks are also extensively used, specially in the earlier years of applying DNN to biosignal processing and decision, as they provide a probability of a classification, accordingly to a number of features in the most diverse types of biosignals [125, 138, 139]. These networks, in resemblance to the autoencoder, are unsupervised methods but usually composed of several restricted Boltzmann machines. These, circulate the information back and forth, trying to replicate the input, but where contrastive divergence is applied to each sub-network in turn [140]. These networks can be used for classification and were easily converted from the human extracted features to traditional ML approaches.

In an interesting work from Gandhi et al. (2015) [141], a network was applied for non-invasive brain-computer interface with EEG. They used the Schrodinger wave equation, producing the Recurrent Quantum Neural Network and a Savitzky–Golay filtering model. This network was designed to efficiently capture the statistical behaviour of the time-series in an unsupervised way and results show that it may potentially be a flexible technique for real-time EEG signal filtering.

Reinforcement Learning (RL) and Generative Adversarial Networks (GAN) are still not prominent in the biosignal studies. One example in using RL in biological signals is provided by Wang et al. (2017) [142], applied to studies with monkeys, using spiking from microelectrodes. In the case of GAN an approach conducted by Osokin et al. (2017) [143] try to extrapolate the synthesis using a latent vector and explored their ability to generate biosignals with certain classes and characteristics.

While reading the literature, it is possible to conclude that most research is made to ECG signals and that CNN are more popular. The accessibility to hundreds of records from of public databases such as Physiobank [144] makes training possible, added to the fact that the ECG has a very clear and known cycle, makes these signals ideal for these technologies to be implemented and tested. As for CNN, the training is faster than the RNN counterparts, while achieving good results.

But in contrast with image and video processing fields, where improvements are achieved at high pace, in the biosignals field, apart from some exceptions, the architectures are still in embryonic development and require further investigation.

4.2 State-of-the-art for the application scenarios

This section covers the state-of-the-art specifically for the three different application scenarios where details can be found in Chapter 8: synthesis of biosignals; ECG biometric systems; and ECG abnormality detection in the context of noise and arrhythmias.

4.2.1 Biosignal Synthesis

The literature states that the most relevant synthetic biosignals are EMG and ECG. Gamboa et al. (2012)[145] simulates this signal using an autoregressive model, in which the parameters were learned from acquired signals, and adding Gaussian noise. The resultant signal was controlled by inputting the noise level and signal power and the instant and duration of the activation [145].

The existent approaches in reproducing EMG include the use of a sum of diphasic waves [146, 147], implement a random EMG tonic wave and multiply by a sinus wave [148]. Regarding the ECG end, various research articles rely on the theoretical expression of this signal. Back in 1989, Murthy and Reddy (1989) [149] describes some simulation methods, such as the combination of cosine waves and the use of Shanks algorithm [149]. The Standard for Cardiac Monitors defined an artificial wave that involves the input of ECG characteristic parameters, such as the QRS amplitude and time, as a standard for designing and validating event detectors [150]. McSharry et al. (2003) [151] created a dynamical model based on three coupled ordinary differential equations where several parameters could also be inserted, such as the mean and standard deviation of the heart rate, the morphology of the cycle and a power spectrum of the RR intervals [151]. Wu and Sengupta (2016)[152] conceived a scalable architecture in silicon that handles several delayed harmonic waves until forming the morphological representation of the ECG [152].

From the pattern detection of acquired and processed ECG, the parameterization of a model is achievable. This mechanism grants the ability to synthesize signals by exploiting the prediction power, using the output as the next input. In this field, several signal processing methods are combined with machine learning techniques. For instance, the features extracted from wavelet frequency analysis may be applied in Dynamic Time

Warping, where each characteristic wave intervals are acquired in the corresponding scales [153]. Hidden Markov Models (HMM) is also used in the wavelet-frequency domain, where this model learns all the coefficients and the signal reconstruction via Bayesian prediction of the next sample [154]. Hilbert-Transform is also applied with the parameters extracted for polynomial approximation of an ECG [155]. Rodríguez et al. (2014) [156] also uses the Hilbert-Transform to remove the highest frequency from the signal, using the reconstructed one to serve as input to train an Artificial Neural Networks (ANN) model [156]. Finally, Atoui, Fayn, and Rubel (2004) [157] uses a multilayer ANN feeding it with raw signals acquired with a 12-Led ECG considering five derivations and establishing a relationship between them [157].

4.2.2 Biometric classification

This section will address recent related work with respect to reference databases on ECG Biometrics, that will be further explored in our work for benchmarking. The Fantasia and MIT-BIH datasets from Physionet are extensively used for biometrics research, while the CYBHi database has been considered by Merone et al. (2017) [158] as one of the two best databases in terms of acquisition protocol and hardware for biometric studies, among fifteen databases. Therefore, appropriate to study a real-life environment. The mentioned databases and corresponding biometric research will be described in detail in Chapter 8.

4.2.2.1 Biometric classification using Fantasia Database

The Fantasia is a database from Physionet which contains ECG recordings of subjects exposed to 120 min of continuous supine resting while watching the Disney movie Fantasia [159].

This database was used by Tantawi, Revett, and Salem (2013) [160], which created a model that uses a reduced version of a set of fiducial features by applying Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), information-gain ratio and rough sets (RS) achieving an accuracy of $90 \pm 8\%$ in PCA, False Rejection Rate (FRR) 5% for LDA and False Negative Rate (FNR) of 4% for RS [160]. The same team published a method that extracts and decomposes R-R intervals using Discrete Wavelet Transform (DWT) and obtained an accuracy of 95.89%, FRR 0% and FNR 5% [161]. The work of Gargiulo et al. (2015) [162] analyses the influence of the HRV in the QT intervals and suggests their correction to improve performance, leading to identification rates with MultiLayer Perceptron (MLP) and Support Vector Machine (SVM) between 97% and 99% [162]. The study described above by Zhang, Zhou, and Zeng (2017) [108] has achieved an accuracy of 97.2% for the Fantasia dataset [108].

4.2.2.2 Biometric classification using MIT-BIH Database

The MIT-BIH Database has been available since 1999 in Physionet [114]. This database has been used not only for validating arrhythmia detectors and other cardiac dynamics

research, but also in the biometrics field. The MIT-BIH Normal Sinus Rhythm dataset was used by Shen et al. (2011) [163], Rabhi and Lachiri (2013) [164] and Sidek, Mai, and Khalil (2014) [165] to validate their models. Shen et al. (2011) [163] proposes a Piecewise Linear Representation for feature extraction and classification using the similarity measures from the Dynamic Time Warping (DTW) method. The results for identification for this method was 100% and the minimum of the half total error (HTER) of 0.2% [166].

The accuracy in Rabhi and Lachiri (2013) [164] reached 99.07%, using a combination of ten features from heartbeat and sixty from coefficients of the Hermite Polynomials Expansion as input to a Hidden Markov Model (HMM) [164].

Several classification methods applied to characteristics of the QRS complex were compared by Sidek, Mai, and Khalil (2014) [165] achieving the accuracy of 98.3% for the Bayes Network, 99.07% for Naive Bayes, 99.07% for Multilayer Perceptron and 99.07% for k-Nearest Neighbor from the same database [165].

A semi-fiducial approach was suggested by Ye, Kumar, and Coimbra (2012) [166] in which a Wavelet Transform was applied together with an Independent Component Analysis (ICA) analysis to detect each heart beat and insert this information together with RR intervals and fed to a SVM classifier. For the selected 23 records with normal sinus rhythm from the Arrhythmia dataset, they reached 86.4% accuracy for subject evaluation, even though the focus was to classify heartbeat classes. Zhang, Zhou, and Zeng (2017) [108] got accuracy of 90.3% for the 47 individuals of MIT-BIH [108].

4.2.2.3 CYBHi

The *Check Your Biosignals Here initiative* (CYBHi) dataset is based on a system that acquired ECG without the used of skin electrodes. In the work of Da Silva et al. (2014) [167] states that it was "devised a data acquisition framework and experimental setup, for large scale data collection from a large group of subjects through an easily repeatable and efficient procedure". Both sensors had a hand-shaped support synchronized via syncPLUX synchronization kit, in which the electrodes were placed on the fingers while the subjects were seated for 2 min in a resting position. The used data consists in long-term sessions, where two acquisitions were made with 3 months difference, including 63 participants, 14 males and 49 females (18–24 years old) [167]. For the purpose of this paper, the long-term sessions from the first acquisition will be named M1 while the second acquisition will be named M2.

The outcome for authentication of this study was an Equal Error rate (EER) of 9.1% using an SVM classifier against the correlation between ECG templates [167]. In another study that used this database, Lourenço, Silva, and Fred (2012) [168], extracted the features from the ECG morphology and used the segmentation of the RR signals using a ANN and obtained with the Euclidean distance classifier an identification rate of 95.2% for the first test and 90.2% for the second test [168]. As for da Silva Luz et al. (2018) [102], described before, archived EER values of: M1 vs M1 of 1.33%; M1 vs M2 12.78%; M2 vs M1 13.93%.

4.2.3 Abnormality detection in ECG

In this work, the objective is to train a network to identify the what is a normal cycle of an ECG signal and classify when a deviation occurs. This classification problem is going to be tested in two different scenarios: noise and arrhythmia detection. In this subsection the state-of-the-art will refer to the use of DNN in both approaches.

4.2.3.1 Noise Detection

A decision rule-based algorithm is proposed by Satija, Ramkumar, and Manikandan (2018) [169] that calculates the maximum absolute amplitude, number of zero crossings and local maximum peak amplitude of the autocorrelation function, after using a modified ensemble empirical mode decomposition. This work was able to discriminate between six signal groups with an accuracy of 98.93%. John, Galloway, and Valys (2018)[170] proposed a 16 layer CNN which predicts once per second on a 10 second inputs. An AUC of 0.977 was reached for this binary classification model with an 88.7% sensitivity. Ansari, Gryak, and Najarian (2018)[120] also developed a CNN to detect usable and unusable ECG segments, in terms of calculating the heart rate variability (HRV). The filtering using the CNN resulted in an area under curve (AUC) of 0.96 for the classification of noise affected segments, compared to an AUC of 0.87 for a Support Vector Machine model.

4.2.3.2 Pathological event detection

More recently, Acharya et al. (2017)[171] used a CNN composed of 11 layers that analyse with 2 or 5 seconds for detection of atrial fibrillation, atrial flutter, and ventricular fibrillation. This study achieved accuracy, sensitivity and specificity values of 92.50%, 98.09% and 93.13% for the 5 second windows, respectively and 94.90%, 99.13% and 81.44% for the 2 seconds windows.

Also, a combination of CNN with RNN was developed by S. Andersen, Peimankar, and Puthusserypady (2018) [131] in order to distinguish between atrial fibrillation and normal segments for ECG recordings of 24 hours. The extracted features using the CNN module were processed by the RNN achieving a sensitivity of 98.98% and a specificity of 96.95%, making predictions for 24 hours of ECG signal in under one second. This algorithm was sensible to noise, reducing significantly its results when present [131].

Recently, Hannun et al. (2019)[119] developed a 34 layer DNN that could detect between 10 different arrhythmias and normal sinus rhythm using raw signal, while also detecting noise corrupted segments, constituting 12 prediction classes in total. The results were higher than the ones obtained by average cardiologists' classification in terms of sensitivity and matching specificity values. The average Area Above the Curve (AUC) was of 0.97 and the confusion matrices were similar, accentuating the same problematic rhythm classes for both [119].

Biosignal Processing for Deep Learning Architectures

This chapter will guide the reader for preparing data for the Deep Neural Networks (DNN) architectures for biosignal processing, decision, classification, and regression, presented in the following chapters. Before describing in detail the several approaches some considerations regarding this type of time-series must be done:

- Normally biosignals have a vast amount of data points, depending on the sampling frequency and time of acquisition;
- They are sensed from living beings, therefore they measure complex systems with semi-randomness that is difficult, or even impossible, to mimic or understand completely;
- The instant (t_n) data points have intrinsic dependencies from the previous (t_{n-k}), as they describe physiological events;
- All biosignals acquired in the same subset at the same time correlate with the others, as the biological systems are interconnected and the acquisition of one biosignal may have traits of the other. E.g. the respiration has a direct correlation with the heart rate and the brain state when emotional responses occur;
- All biosignals are prone to noise, both from internal or external sources.

All of these items are what differentiate the technical issues from image, video, and text-processing commonly used in the DNN research. Therefore for the sake of feeding the signals into the proposed architectures, they need to be submitted to a sequential approach where the signal is first preprocessed.

5.1 Data Processing

The data preparation stage allows the networks to learn the morphological structure and mechanics of the selected biosignal. Recalling that the DNN learn a function based on input, they should only access what they are supposed to learn, while discarding what is not deemed necessary. Some of the proposed methods to use on the raw data processing are presented in this section.

5.1.1 Hanning Window Moving Average

To remove the higher noise frequencies of a signal from the time-domain, which often is not representative of the physiological traits to be learned, the smooth method may be applied. The transformed signal will be a soft version of the original (x) and can be obtained by convolving it with a Hanning window (H). Each value of the discrete Hanning window (H_m) may be described mathematically as:

$$H_m = \frac{1}{2} \cdot \left[1 - \cos \left(\frac{2\pi n}{W-1} \right) \right] \quad (5.1)$$

where m and W are the window's index and size, respectively. The vector s , with size N , is calculated with the convolution of x with H :

$$s_n = \sum_{m=0}^W a_n H_{n-m} \quad (5.2)$$

where s_n is the n -th element of the vector s . Therefore each sample s_n is the sum of the product of each point of the Hanning window with the raw signal.

5.1.2 Normalizing

The normalization step is very important due to two reasons: data uniformization and scaling for the networks to absorb the information. It consists of altering the values of the signal to fit a determined scale, while maintaining its morphology, and allows the networks to learn an abstraction of their characteristics. There are several ways of normalizing the signal (x), but only three will be presented:

$$\bar{x} = \frac{x}{\max(|x|)} \quad (5.3)$$

where the normalized signal (\bar{x}) is the composition of ratio between the maximum of the modulus ($\max(|x|)$) and x . This equation is the most simple and naive, as it works if the signal does not have artifacts or if it is applied to single windows. This method may be applied to a sliding window:

$$\bar{x}_n = \frac{x_n}{\max(|x_{[n-W/2:n+W/2]}|)} \quad (5.4)$$

where \bar{x}_n is the n -th normalized sample of the time-series and $|x_{[n-W/2:n+W/2]}|$ is the modulus of the maximum of the surrounding window. This method must be used at most discretion, as it can corrupt all signal if the W is not well selected. It can only be used in a cyclic signal and each window must contain at least one cycle, but it is advisable that it has more than one. It can be very useful when the signal has several artifacts reducing the variability between signals in this conditions.

$$\bar{x} = \frac{x}{\max(x) - \min(x)} \quad (5.5)$$

this case, also named "min-max normalization", keeps the maximum amplitude of the signal of 1. It can be used to help the classification methods to perform better. This can be made in the windowed variant as well, with the same precautions.

$$\bar{x} = \frac{x - \min(x)}{\max(|x - \min(x)|)} \quad (5.6)$$

this normalization removes the negative values, but also removes the mean invariance of the signal. Apart for being a requirement for the quantization step, it may also be used to help a Convolution Neural Networks (CNN) network to converge. As stated before, the sliding window may also be applied.

$$\bar{x} = \frac{x}{\text{std}(x)} \quad (5.7)$$

where std is the standard deviation. While using the sliding window this normalization dampens the areas where there is higher changes in amplitude.

5.1.3 Subsampling

This technique is used for data compression and to reduce the computational burden, although it can reduce the amount of information and sometimes distort the signal. On this account, a compromise must be made and it is advised to use when the number of data points is too high for the computational power of the used Central Processing Unit (CPU), Graphics Processing Unit (GPU) and/or Random Access Memory (RAM). The advised method in favor of reducing the information loss would be decimation, as it prevents signal aliasing. Before reducing the sampling rate it applies a low pass finite impulse response (FIR) order 8 Chebyshev type I filter, removing all frequencies above the Nyquist frequency.

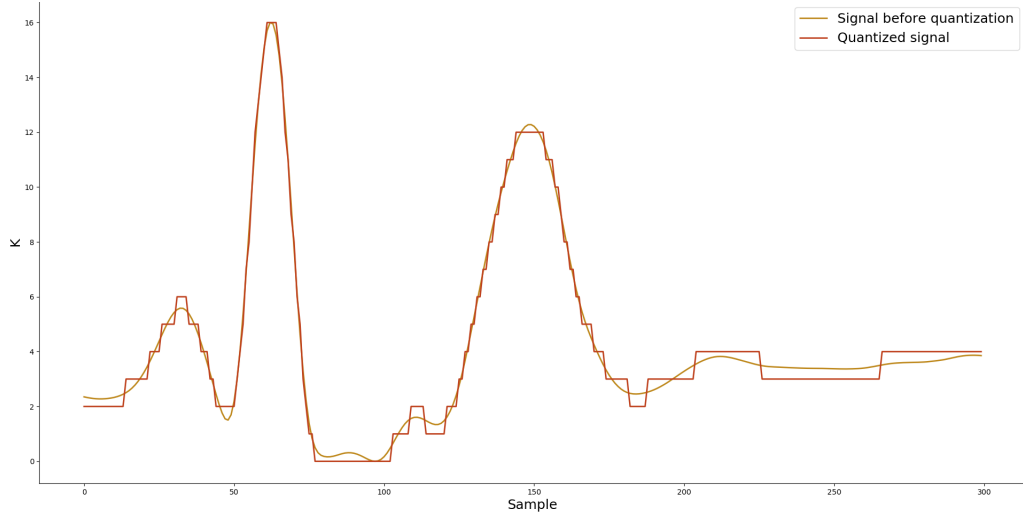


Figure 5.1: Example of an ECG transformed by a quantization process with a signal dimension (S_D) of 16, where K represents the amplitude.

5.1.4 Quantization

The quantization is made when the network needs a discrete representation of the signal to help it process the provided sequence. The amplitude of each sample n will be given integer value, after the normalization step done by equation 5.6 the input signal dimension is decreased with the following equation:

$$\hat{x} = \lceil \bar{x} \rceil \cdot (S_D - 1) \quad (5.8)$$

where $\lceil \bar{x} \rceil$ is the nearest integer function, also named *round*, applied to the normalized signal. This will be multiplied by $S_D - 1$, where S_D is the signal dimension parameter. Consequently, \hat{x} will be a vector where each position will have an associated step k that corresponds to the reduced dimensional shape of the signal value, where $k = \{0, 1, \dots, S_D - 1\}$ as depicted in Fig. 5.1

5.1.5 Segmentation

This procedure is crucial for the networks process the signal, one could say that a signal window may be translated as a phrase of the biosignals. Therefore each segment must contain the relevant traits that we want the network to learn. This procedure consists of separating the signals in time windows (TWs), i.e. fixed-size intervals, that are directly fed to the model with the possibility of window overlapping. The possibility of sharing information in every window should be beneficial because the signal can be seen under multiple phases of the signal sharing its continuity with the network. By increasing the percentage of overlapping, it is possible also to increase the data to be given to the network when insufficient, although at some point it can turn out to be redundant.

5.1.6 Spectrograms

A spectrogram is a representation that constitutes the frequency spectrum of a signal along time. The x-axis represents the time, while the y-axis represents the frequency. The energy of each frequency at a given moment in time is usually a third dimension, either a color scale or values in the z-axis, creating a 3-Dimensional surface. The frequencies and their magnitudes can be obtained by the application of a Fourier transform to successive signal windows.

5.1.7 Noise Segments Elimination

When the signal is too corrupt to be used, the best approach is to remove it. As this information is not needed to learn the clean characteristics of the signal, artifacts may also have a great impact on the normalization techniques, invalidating all signals for analysis. The classification and decision methods, except when searching for the noise, can be highly affected by these artifacts. Therefore it is advisable to use two methods:

- The first is removing the windows that are between intervals around a percentage of the median of the standard deviation value and the median of the mean value of each window, depending on the database noise factor;
- The second method is applying the algorithm developed at the LIBPhys biosignals group, by Rodrigues, Belo, and Gamboa (2017)[11] when the first method does not work. This algorithm extract features that best characterize the shape and behavior of the signal over time and clusters each group of samples using the agglomerative clustering approach. After testing in numerous datasets this algorithm was even able to successfully detect different types of noise. Even if this method was tested in ECG signals, it can be extrapolated for other types of signals. Useful features can be the standard deviation, absolute mean and zero-crossing rate.

5.1.8 Symbolic Search in Time Series

The Syntactic Search in Time Series (SSTS) [172]¹ as a solution for exploring and search patterns in time-series. This is an interactive tool that relies on expressing the morphological reasoning based on the visual interpretation of patterns in time series and is comprised of three steps:

1. Pre-Processing: the transformation of the signal using typical processing methods;
2. Symbolic Connotation: translating the time-series into a text with symbols representing the properties that help on the search of the target patterns;
3. Search: Phase where the connotation is exploited and a string-based search is used to find the requested pattern, by using regular expressions.

¹developed at the LIBPhys biosignals group, with a partnership with Fraunhofer AICOS Portugal

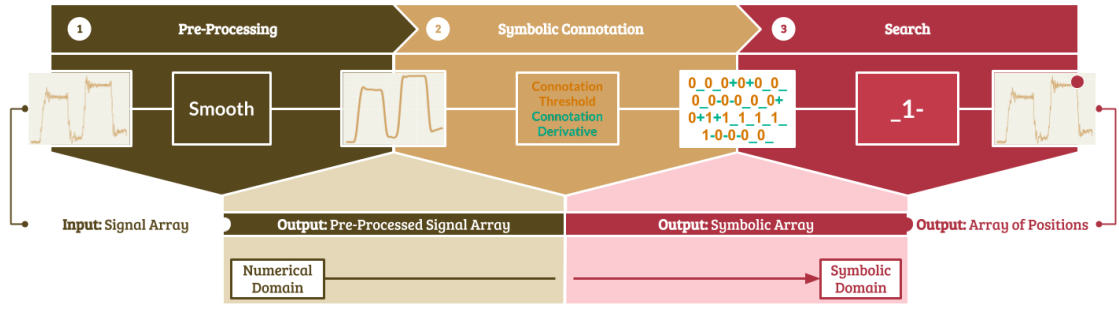


Figure 5.2: The SSTS is divided into three main modules - pre-processing, symbolic connotation and search. From [172].

This tool is an evolving mechanism to augment the cognitive expressiveness of the search while increasing the abstraction over solving time-series query tasks. In the context of DNN applied to biosignals it can be very helpful for labeling the signals when searching for a specific event (e.g. activation of a muscle using Electromyogram (EMG)) in a semi-autonomous way, helping the physician or data-scientist to label events for supervised learning.

It can also be used to be used for unsupervised learning while generating signals. By using the transformation capability of the connotation system, which decreases the signal dimension while attaining the most important information of the signal that the user wishes to retain, can be especially useful for an agnostic signal generation that will be explained in more detail in section 7.4.2.

Architectural Modules

This chapter will present the modules that build Deep Learning (DL) architectures specific for signal processing and decision presented in Chapter 7. Fig. 6.1 depicts the various diagrams and symbolic representation that will be used throughout chapter 7. The different exposed modules are described as follows:

1. Convolution Neural Networks (CNN) encoder - extracts features based on the required output with a convolutional network approach;
2. CNN decoder - generator of a biosignal from a latent space with a convolutional approach;
3. Recurrent Neural Networks (RNN) encoder - extracts features based on the required output with a recurrent network approach, this approach is usually constructed in a structure of many to one;
4. RNN decoder - generator of a biosignal from a latent space with a recurrent network approach, the many to many architectures can work as both encoder and decoder;
5. Latent vector - feature vector that contains the information for the output to decipher. This vector contains the encoded information about an input, which can be translated by the output module. Usually the encoder module makes this vector by extracting machine-learned features. It can also be manipulated for a generation;
6. Dense Network - a fully connected network, i.e. a classifier that may be used to to translate from the latent vector to the desired output;

Several configurations may be brought upon the combination of several modules. For example, when a network comprises a CNN encoder, a latent layer, and a dense network,

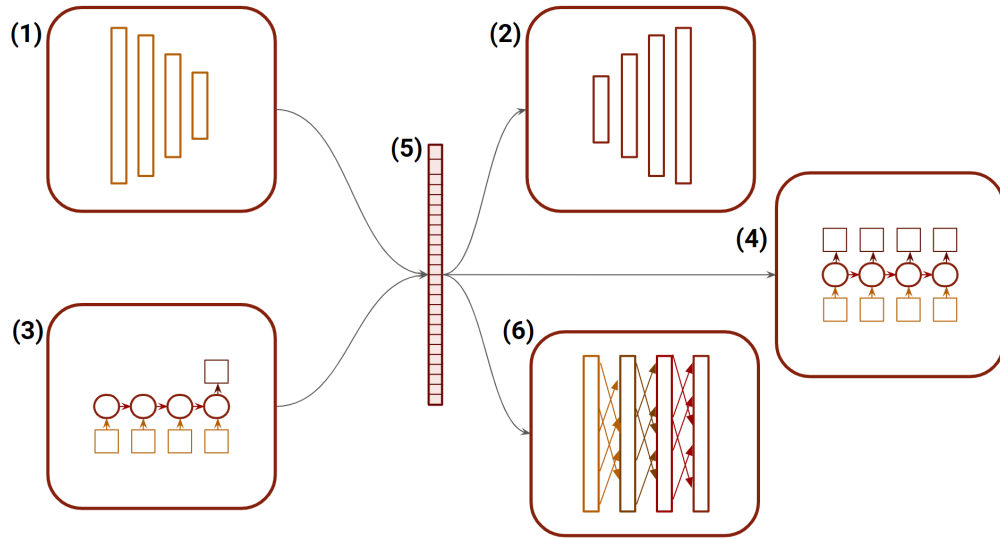


Figure 6.1: Diagram of the deep learning configurations modules for biosignal processing and decision. 1. CNN encoder; 2. CNN decoder; 3. RNN encoder; 4. RNN decoder; 5. Latent vector; 6. Dense Network.

then a classifier is formed, where the CNN automatically learns the required features for the classification. Another example is the CNN encoder and RNN decoder, it can represent a generator based on the extracted features of the encoding part. This is a Recurrent CNN to be used, for example, for the anomaly detection segment where the features that code for the normal segments are different from the abnormal settings.

6.1 CNN encoder and decoder

The encoder may be simple as containing several stacked layers with decreasing number of input samples. The number of features may be smaller, the same or higher than the input depending on the combination between the stride, size and number of filters, the size of dilated convolution and the padding type.

6.1.1 Filters and Stride

The stride is how many samples the convolution skips. The higher the stride less points will be considered between convolutions. The weights, also named as filters, has a size equal to the number of samples, it can be a vector (1D) or matrix (2D), therefore how many points are computed in each convolution.

The dilated convolution is the gaps between the points that each filter will consider, e.g. when it has dilation of 1 then the convolution is normal, when the value is 2 then there is a sample gap between each point of the sample that will be analyzed. This method

is a way of increasing the global view of the network with less cost of computation and it is similar to subsampling.

Fig. 6.2 depicts five examples of one filter configurations. The first case, the convolutions are almost all juxtaposed, because the stride is 1 since there is a computation per sample, then the output will have the same size as the input, in the second and third example, the size will be reduced by half, as the samples are skipped by two each convolution. The dilated convolutions, displayed in cases 4 and 5, are an example of reduced dimensionality, being the size the dilation is the reducing factor. The number of filters dictates the multiplying factor in the output. For example, if the time window has 42 samples, the stride of 2 and 3 filters, the final output will be a matrix of size (31×3) , totaling 93 features. Pooling can be also used to reduce the signal temporal dimension at the end of each or a block of convolutions.

6.1.2 Sequential Concern

The convolutional network can be made using several sequential layers, as the depth increases, the capability of memory is higher and the more specific the features become. But unfortunately, it increases the computation time, and the vanishing gradient may pose a problem. If this happens then it is advisable to add residual connections between blocks of layers. For example, one could use the topology of DenseNet from Huang, Liu, and Weinberger (2016) [173] which block receives as input all the features from the previous blocks.

While planning this architecture, at the end of each step, one should consider the use of techniques that stabilize the training of the network, such as batch normalization and hyperbolic tangent activation in the last convolutional layer. The batch normalization layer increases the stability of a neural network by subtracting the mean and dividing the batch standard deviation.

6.1.3 Decoder

The decoder may use the inverted convolutional matrices, but it is advisable to have a dense network (with one or two layers) in the end with many hidden units the size of the desired output. Normal convolutions also work, but the number of features must be controlled as they pass from layer to layer, as they should have the same or more features than the desired output in the last convolutional layer to ensure all the information is generated.

6.1.4 Pros and cons

This architecture poses some limitations, as the CNN does not relate temporal dependencies between samples within the sequences, but just treats each vector as a whole time-stamp. Therefore, even if it helps in terms of classification, encoding, and decoding time-windows,

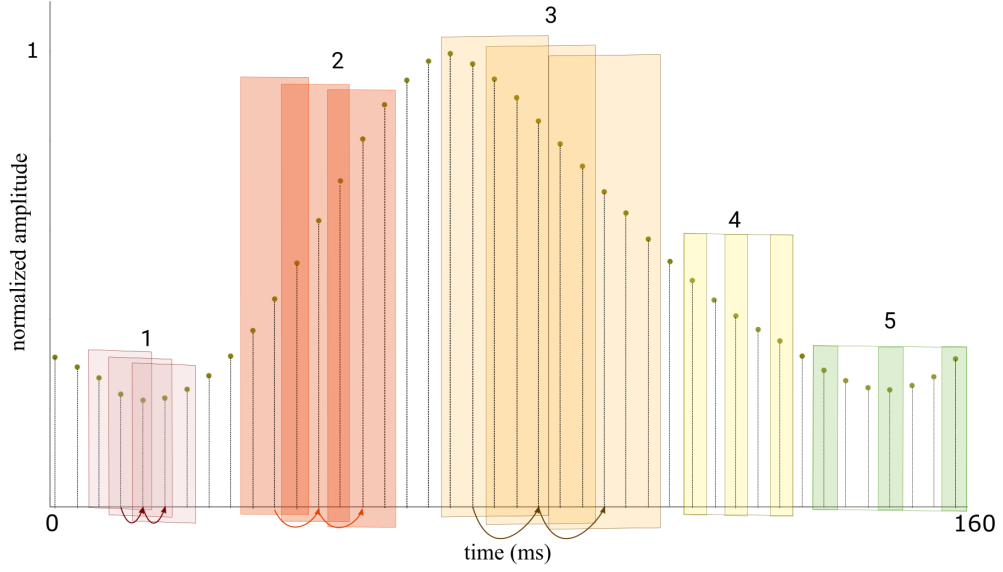


Figure 6.2: Time window of a respiration signal with 42 samples and examples of filter configurations: 1. Size 3 filter with stride of 1; 2. Size 3 filter with stride of 2; 3. Size 6 filter with stride of 2; 4. Size 3 filter with dilated convolution of 2; 5. Size 3 filter with dilated convolution of 3;

it is not suitable for prediction or in generating higher dimension sequences, which makes RNN more appropriate.

6.2 RNN decoder and encoder

The RNN, namely Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) can be stacked in the number of units, as they increase the memory of the network. The number of units increases significantly the number of the dependent computations, and therefore the network increases in terms of computational complexity. These networks have the limitation of not being able to access all the Graphics Processing Unit (GPU) faculties, as the computation dependencies from cell to cell (in a multi-layered architecture) and state to state (in terms of sequence samples) cannot be parallelized.

6.2.1 Signal Quantization

There is a consideration that should be used while learning from these networks, due to their gated nature, the output should be a vector with the probability density function of each class. To explain this concept, one example of a generation of natural language processing will be given, where each character is represented as a "one hot" vector. This vector has the length of the vocabulary, i.e. the number of possible characters and is sparse where only the position of each character has one [174]. Therefore, the softmax layer before each output gives the probability of each position is the one. By extrapolation of this

concept when considering a character as a point in a signal vector, and each segment as a sentence. This can be achieved by quantizing the signal, and each integer value represents each position of this vector.

6.2.2 Input transformation

The use of one embedded matrix (E) could help the performance of the training sequence, it has of the size of $(S_D \times H_D)$, where S_D is the signal dimension, i.e. the size of quantization, and H_D the hidden dimension, i.e. the size of the hidden units on the first cell. Hence containing all representation vectors for each possible sample value and the transformation is simply $\hat{x}_n = E_{[x_n]}$ considering that x_n is an integer scalar, the $E_{[x_n]}$ represents the x_n -th column vector of E and \hat{x}_n will be the vector that will enter the first RNN cell. This matrix is used as a dictionary that gives an image vector \hat{x}_n of the scalar x_n and is also a training parameter, adjusting itself for better representation in each iteration [175].

The input to the network could also be features engineered by convolutional layers, but to fit each RNN cell dimension, a connective dense layer should be included between the output of the convolutional network and the input of the first cell.

6.2.3 Encoding vs Decoding

When considering the encoding and decoding of the signal, one should think that the main difference is the many-to-one approach for encoding, i.e. ignoring the output of each time-step, and using only the last output. Also, another approach is to generate the latent space as the concatenation of all outputs, but the increasing computational time has to be considered. For decoding or generating a signal, all the outputs should be considered.

6.2.4 Pros and Cons

The RNN have a good representation of time-sequences and therefore an interesting choice for prediction and generation, but due to the computational requirements and training difficulties, the usual approach of classification is not completely advisable. The CNN networks could offer the same results in less computational time.

6.3 Dense Network

The dense networks are mostly used for the transformation of the input or classification. Only one layer with a linear activation function could be enough for fitting dimensions from one input to a specific output in different cases, such as:

- From the output of a layer to the input of another layer with different dimension;
- From the latent vector to input of a layer;
- From the layer to the output;

- From the output of a layer to a signal or sample generation;

One can also use these dense networks with more than one layer to increase the learning of non-linear relations, especially in the case of classification. It is common to have one layer with a large number of hidden units (for example 1024) before the classification layer. The output of the last classification layer is commonly named as logits, where the activation is linear. Including the softmax activation function on top of the logit, the vector will give the probability density function, where each position of the vector will give an estimate of the corresponding class or value.

6.4 Latent space and fusion layers

As mentioned earlier, the latent space may be a composition of the features extracted by the mentioned encoding modules but it also could be human-engineered to generate a controlled output. For example, the variational autoencoders are a branch of generators that learn the statistical distribution of the input and generate the output based on the training set. Therefore, as one can manipulate the input of that distribution, the network may produce new data generations [176]. In the case of signal processing, one could use the information of the morphology and insert directly into a generator, reproducing a signal, by using the R information of an Electrocardiogram (ECG) or using a Syntactic Search in Time-Series (SSTS) connotation representation.

Several latent vectors may be combined, in what is called a fusion layer. This layer is a concatenation of the several modules or a simple mathematical rule, such as sum, averaged or even multiplied. The combination of input encoders may provide a higher information spectrum needed for a classifier to have better results.

Several configurations of the modules with the latent space in the middle can change dramatically what the output will be, as different paradigms are needed for each problem in biosignal engineering. The next sections will provide a detailed explanation of each configuration and in Chapter 8 the stated three examples of application in the areas of synthesis, biometry, and anomaly detection.

Neural Architectures for Biosignal Processing and Decision

The last chapter states the different modules which can create new architectures. These architectures specific for signal processing, decision and generation will be depicted in this chapter.

7.1 Supervised Learning

This is the most common configuration for supervised learning. The convolutional network is supported by the parallelism of the Graphics Processing Unit (GPU) computation making these networks faster to train than the Recurrent Neural Networks (RNN) counterparts. Filters will be learned in these configurations when using a 1D CNN. When fed with a time-windows with a portion of the signal, filters will activate if different morphological shapes are detected. When the convolutional encoder is combined with a dense module which allows the weights to map between the input and a class output, the created latent vector will be specific for the problem at hand.

7.1.1 Multimodality approaches

When the detection could be enhanced by observing patterns from several sources, such as scales of the signal, channels or frequencies, there are two main approaches that one could follow, like depicted Fig. 7.2. The configuration displayed in Fig. 7.2(a) refers to the implementation of several 1-dimensional CNN for each channel and joining them into a single fusion layer that combines the different extracted features into one vector. The Fig. 7.2(b) architecture makes use of the capacities of the 2-dimensional CNN to detect similarities between the near channels.

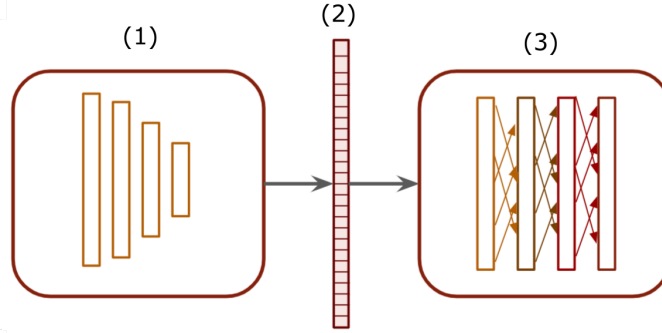


Figure 7.1: Diagram of a CNN classifier. (1) CNN encoder that extracts features in a time-window. (2) Resultant latent vector. (3) Feedforward network that classifies the time-window.

Both these systems have downsides. By understanding the nature behind the 2D CNN architectures, one realizes that they were created to detect patterns in images in both horizontal and vertical axis neighborhood pixels, where the information is related. In the case of biosignals, this must be considered. For example, exploring frequencies for each time-stamp may have its upsides because the spectrograms are images, therefore a close relationship can be found, but for multimodal and multi-channel this may not be the best approach. The Electroencephalogram (EEG), Electrocardiogram (ECG) and Electromyogram (EMG) signals with various channels could be used for this architecture if the sensors were close to each other during acquisition and if they are in the same sequential order. This approach also does not work when one channel is missing, unless the network is trained for such an event. In the case of a multi-model, the biosignals should have a close correlation, such as ECG and Blood Volume Pressure (BVP).

The fusion network works for the channel configuration and, depending on how the training is made, they can be independent of each other. For example, if a model is pre-trained using autoencoders for all biosignals (see Section 7.2) they become prepared for the features from each channel. When the model is reused in parallel, they can behave independently. Depending on the type of the used fusion and the order of the channels have different impacts. If concatenation is used, then the order has a major significance, and signals such as EMG where the activated muscle has a major impact on the outcome could benefit from it. On the other hand, if the sum is the rule for fusion, then the order does not matter, and signals such as EEG of the same area of the cortex would make training simpler and faster. Using an averaging one, the impact of missing channels would be lessened.

Both configurations have been used in the examples of section 8.3.2, by using 2D spectrometry and using a 1D convolutional network with a fusion layer.

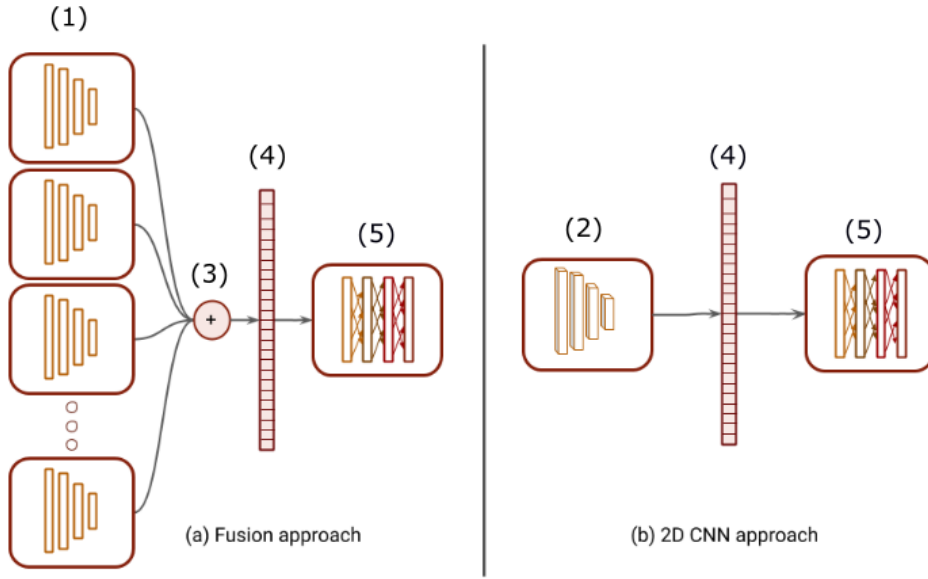


Figure 7.2: Diagram of two CNN classifiers for multi-channel biosignal approaches. In the left, (a) Encompasses the integration of several CNN networks with fusion layers; (b), in the right, depicts the integration of 2D CNN. (1) Array of CNN encoders that extracts features for each channel. (2) 2D CNN encoder. (3) Fusion layer that connects all the outputs of all array encoders. (4) Latent vector which includes the extracted features. (5) Feedforward network that classifies the sequence.

7.1.2 Recurrent classification

The configuration for classification can be a many-to-one approach (Fig 7.3), where only the last state is required for this to be achieved. The complexity of the classification can be increased by adding all the states and/or the other RNN layers if they are implemented with more than one cell in each state. This architecture can be utilized with all biosignals, as they exploit the mechanics behind the evolution of the signal. Theoretically, as the RNN has good prediction capabilities, this configuration could be used to classify events that have not occurred. Unfortunately, these networks are known for the difficulty of converging, therefore preprocessing techniques such as quantization and decimation or increasing the hidden units and trying different weight initialization methods is advisable.

To increase the number of samples covered by the classification method it is possible to combine a reusable CNN encoder for each input of the RNN. This way the CNN could extract features from several windows and stack them in a latent sequence, facilitating the RNN classification (Fig. 7.4). If the encoding is not too complex, this architecture could ensure the reduction of the computational effort, as the required sequence used in the RNN part will be shorter, compared using only the sample directly in the RNN. To create an interface between the CNN encoder and the RNN classifier, one should use dense layers because of two reasons: give weights to each feature and compatibility between the latent tensor dimensions.

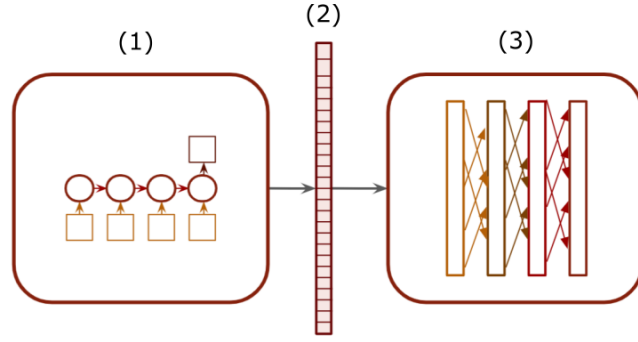


Figure 7.3: Diagram for classification using a simple RNN configuration. (1) RNN encoder that extracts features in a sequence of windows. (2) Resultant latent vector. (3) Feedforward network that classifies the sequence.

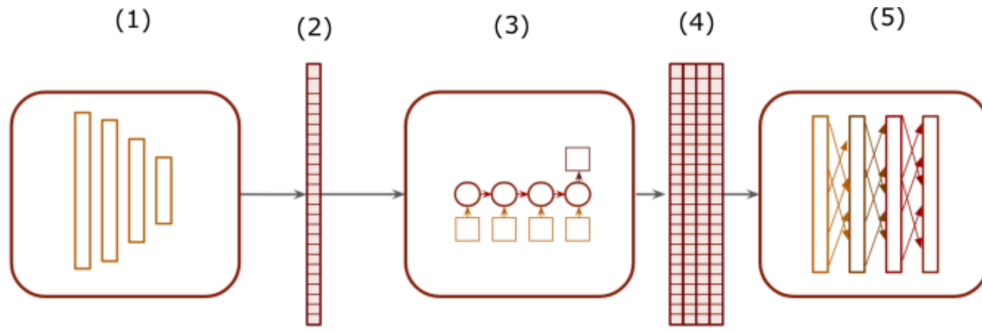


Figure 7.4: Diagram for encoded sequences classification using RNN architecture. (1) CNN encoder for each time-window for feature extraction. (2) latent vector for each time-window. (3) RNN encoder for a sequence of latent vectors. (4) Stack of latent vectors from sequential time-windows. (5) Feed forward network for classification.

7.1.3 Biosignal and time-window considerations

Even though these configurations may be used for any kind of biosignals some considerations must be made. One of them is that each sequence must contain information on what is supposed to classify. If it is not possible to portrair this information in any configuration due to computational limitations then downsampling is advisable. For example, in ECG to have an R peak detector, then the window should be longer than the size of a QRS complex, on average between 80 ms to 120 ms, but to detect an asymptomatic event, at least a cycle of 1 sec should be considered. In the case of classifying motor control using EMG, one should consider the intervals of time required for each activation and desired detail.

7.2 Unsupervised feature extraction with autoencoders

Autoencoders are a useful architecture for pretraining a network, as it learns the basic structure of a signal and later uses the trained encoder for other purposes. A rhythmic signal is more adapted for this process, for example, the ECG, that has a clear cyclic behavior, or an EMG that can have different orders of activation are two examples where this architecture works.

One important aspect of this type of architecture is the overlapping method and the need to look at the signal at different time windows. When the training is made with a window that is always centered at a component (for example the QRS complex of a ECG) the generator learns how to mimic the central area, without the need for the encoder to extract the required features to represent it, behaving independently.

After learning, the generator part may be discarded and substituted by a dense module or a RNN for example. For RNN prediction, the generator may be used at the end of the cycle to predict the outcome after a sequence of iterations. These architectures are also good for transfer learning, as described in the next subsection.

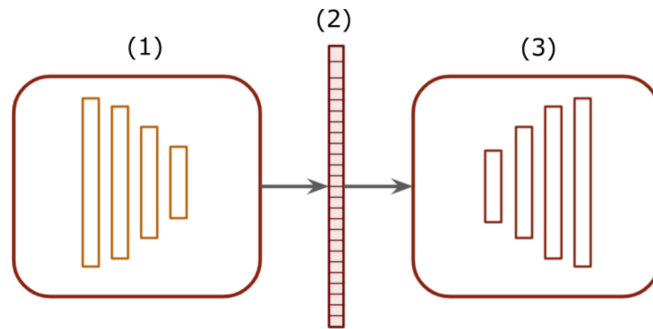


Figure 7.5: Diagram of an autoencoder for feature extraction. (1) CNN encoder that extracts features. (2) Resultant latent vector. (3) CNN decoder that recreates a signal from the encoded information from the latent vector.

7.2.1 Transfer Learning

Transfer learning is the knowledge transference between different domains. It aims to leverage knowledge from a source domain to improve the learning performance or minimize the required labeled examples in the target domain. The closeness between source and target data affects positively the performance of the learning algorithm. This methodology is based on the replacement of the last classification layers with new layers, while locking the training of the encoding layers or fine-tune them with a low learning rate during training. Transfer learning helps to apply Deep Learning (DL) algorithms in real-life scenarios, specially when the ideal scenario of abundant labeled data with the same distribution of classes in both training and test datasets is not possible [177].

Transfer learning is a popular method used in Deep Neural Networks (DNN). In computer vision and natural language processing, this reduces the computational time required to develop these networks, providing a great jump in terms of performance. In the case of biosignals, there is a vast amount of anomalous behaviors and a classification method for one anomaly may not be enough to be used in another model for a completely different anomaly. But by understanding the basic concepts of a signal using an autoencoder and fine-tuning it to other problems may be a simpler and faster approach. This is also interesting when dealing with small databases. For example, when dealing with low amounts of data, the network may not converge to the optimal solution. Therefore one could reuse an autoencoder architecture training with a higher amount of data and after training it, it could be retrained with a smaller database.

An example of this method is depicted in Fig. 7.6 where a CNN with 4 layers for encoding and three layers used for decoding learned how to reproduce a ECG signal. The first dataset is the Fantasia dataset [159], which has 40 recordings with 60 min, and the second is a dataset with 18 recordings of 5 min, both of them are normal ECG and have a sampling rate of 250Hz, both of them were normalized and decimated to a factor of 4 (62.5Hz). Due to the low number of windows of the second dataset, to train an autoencoder, the first was used to train first. When the training was finished and the model was submitted to the second database, during the first epoch, the recreation of the signal was not random and it could reproduce some morphological features. After approximately 100 epochs the training ended and the new signals were successfully recreated.

7.2.2 Encoding sequences

In this case, an autoencoder can be used to learn to recreate time-windows, and later the latent representation may be stacked with other latent vectors of the same sequence. This allows the feature extractor of the CNN encoder to extract the important information of the short-time dependencies, while the RNN encoder can encode the sequence for higher time-dependency information. The latent layer should pass through a Dense layer before entering the RNN for compatibility between dimensions.

7.3 Semi-supervised Learning

By combining these pre-learned features using autoencoders, if the generation layer is substituted by a classification layer one could use it to classify an abnormality.

7.3.1 Classification with pre-learned features

Using the encoder architecture, one can use it to fine-tune the classification at hand, or for direct classification. The encoding of sequences can be also be used by pretraining the RNN to encode information in the last state. This method can also be used for classification.

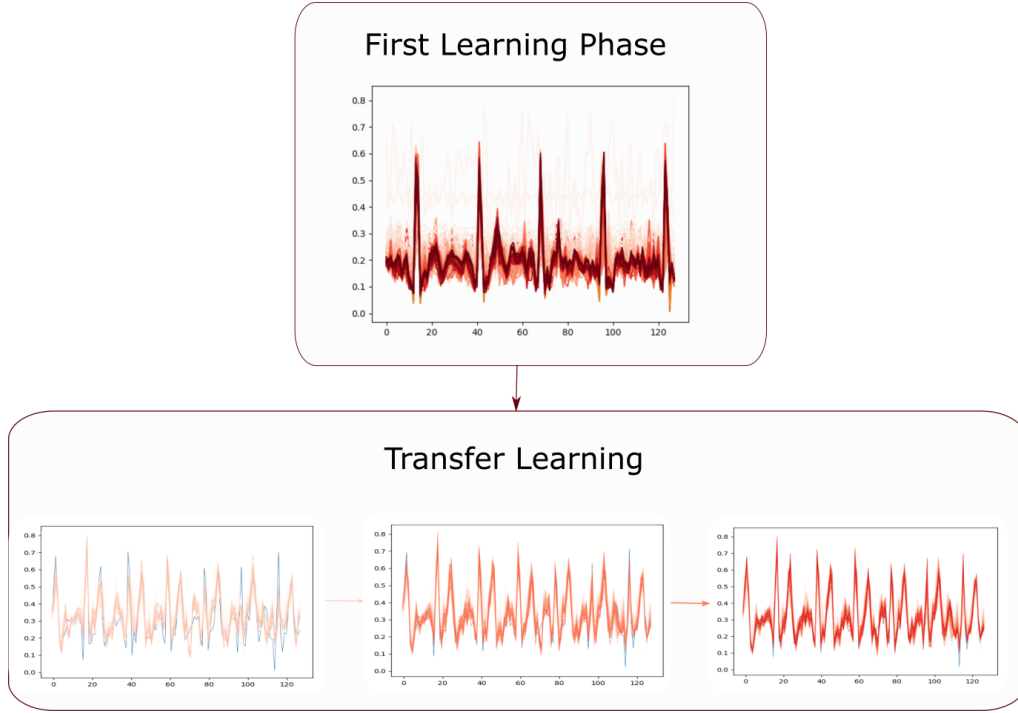


Figure 7.6: Example of learning of a CNN autoencoder used for transfer learning, where the color gradient represents each iteration during training. The darker the color, the higher the number of epochs. The first learning phase used the Fantasia dataset containing 40 individuals with the first hour of data. The second phase was a database containing 18 records of 5min each during three stages of transfer learning.

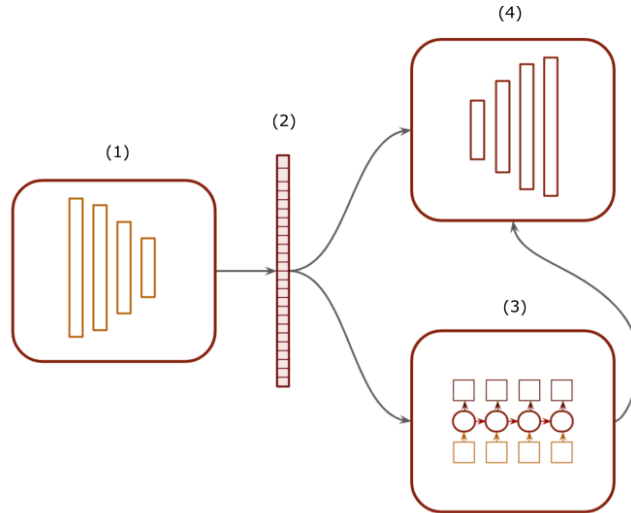


Figure 7.7: Diagram for classification using a simple RNN configuration. (1) CNN encoder for feature extraction of each time-window. (2) Latent vector incorporating encoding of the time-window. (3) RNN decoder which predicts the last window in a sequence. (4) CNN decoder that reconstructs a signal from a latent representation.

Both these mechanics are presented in Fig. 7.8. Both these architectures were used in the classification of abnormal events as seen in Section 8.4

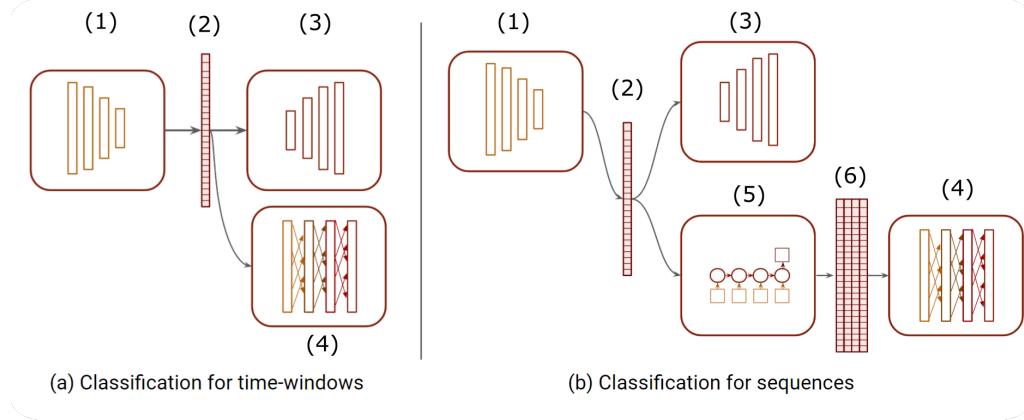


Figure 7.8: Diagrams of classifiers using previous learned feature extraction. In (a) the classification using a single time-window; (b) classification of a sequence of latent vectors using RNN. (1) CNN encoder for each time-window. (2) latent vector for each time-window. (3) CNN decoder for time-window reconstruction. (4) Feed forward network for classification. (5) RNN encoder for a sequence of latent vectors. (6) Stack of latent vectors from sequential time-windows.

7.3.2 Active Learning

Active learning is a method that inquires interactively the user, also named teacher, to label examples with desired outputs. This is mostly used for unlabeled data which is expensive to label. Therefore, continuous learning of a network is a challenging subject, as the learner needs to understand when new classes are given and distinguishing them with the previous known ones.

The adaptation to new classes or commands without the complete retraining of a network may be done by using clustering methods. In this section, a theoretical approach will be presented that could enable this automatic mechanism, depicted in Fig. 7.9. To produce this effect in this proposal a list of requirements must be met:

- A maximum number of classes/commands must be specified;
- A labeled dataset containing all the classes;
- A threshold of confidence for classification;
- An extra class for the unknown needs to exist;
- A user feedback must be provided.

In this approach, two different networks will work in parallel, one for classification and one that uses the sequences to detect abnormal behavior and decide if a new label should

be created or not. The classification layer is composed of a CNN encoder and a dense layer with the capability to classify more classes than exists in the first dataset. A threshold of confidence will be required, in which a known class will only be attributed if that value is surpassed after the softmax layer. Values below will be given to the "unknown" label.

At the same time-frame, each latent vector produced by the encoder will be stacked and a RNN network classifier will be used to ensure that the progression is normal. Consequently, this network will have two actions, "no_change" and "create_label". Theoretically, if this network detects an abnormal behavior when the extracted features are close to each other, it will classify as "create_label", prompting the user if a new class is to be created. The result of the user will reward the system accordingly if it was intended than the system will have a positive reward, if not the system will be rewarded with a negative value.

At the same time, if the "create_label" is selected, then a new label is created in one of the unused positions of the softmax layer of the dense layer. The user will be asked to repeat the biosignal signature several times to be included in the labeled dataset. If not, then the last windows will be labeled as the new class. This empowered dataset will be used to retrain the classifier network, ensuring that the encoder will fine-tune the feature extraction to the new set of classes.

An example would be a network using a EMG-based movement detector to control a prosthetic device. As new patterns of muscle contraction of an arm arise from the user intent, new movements could be created. The user-machine interface would adjust itself to the user, as the user adjusts himself to the machine.

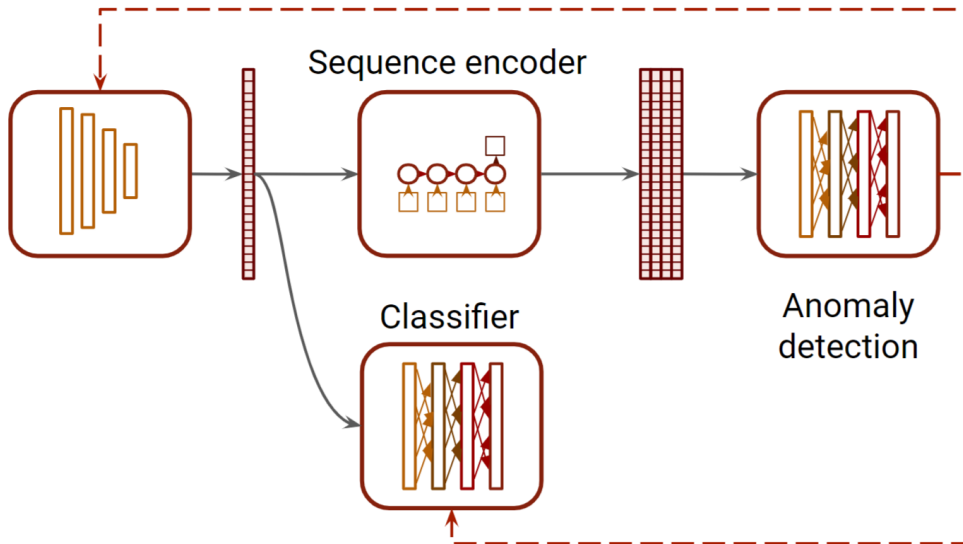


Figure 7.9: Diagram of an reinforced learning mechanism for active learning. The anomaly detection network is a classifier for normal patterns. If an abnormal pattern is recognized, a new learning phase is initiated with a new label. The classifier is a mutating entity that grows as new labels are created.

7.4 Biosignal Generation

In the medical field, data must be handled carefully mainly due to privacy issues and accessibility from the outside of the acquisition facility may prove impossible. With this in mind, generators may be an interesting method to create higher amounts of data that would be used to train other Machine Learning (ML) models, mitigating this issue.

This section will present some architectures that could be proved useful in learning and recreating the biosignal mechanics.

7.4.1 RNN Generators

Encoding sequences may be made by using each sample and trying to reproduce the next sample, without the need to extract features. By using the many-to-many configuration of a RNN one can predict the next sample based on the previous ones in a sequence. These generators only work when a certain pattern is to be expected, for example, the repetitive motion of the ECG, BVP, and Respiration Signal (RESP) or of the Accelerometer (ACC) and EMG during the execution of repetitive movements, such as running and biking.

The RNN encoder can learn the cyclic behavior of a wave and reproduce it accordingly. For this architecture to learn the signal it should be quantized first as it does not absorb the information of the "continuous" signal. An example of the signal generation can be seen in section 8.2

7.4.2 Agnostic Generators

The generator only needs some markers of the original signal to learn how the input is related to the output. For example in the case of ECG, one could only give the information of the R locations and the system would try to reproduce the signal based on those maps. Using the mean-squared error one can reproduce the signal where the R peak is encoded.

From experiments, the results suggested that in the case of a simple CNN generator with 3 layers and a final dense layer it can produce synthetic ECG. The synthetic ECG, presented in Fig. 7.10 has a simple morphology created by geometrical constraints. In this specific case, this architecture can understand and recreate the signal only from the information on the R peak location. But when applying to real ECG, only the examples of the training set can be reproduced correctly, as most of the testing set was not able to be produced, i.e. the RR segments that were not seen previously by the network weren't able to reproduce the respective signal correctly. This suggests that for a generalization of the ECG signal other strategies such as data augmentation, the use of RNN or even Generative Adversarial Networks (GAN) architecture have to be implemented. The examples depicted in Fig. 7.11 show the output for two different individuals from Fantasia dataset (details of this dataset in Chapter 8) of windows of the training set and testing set. The input was the R output, where the places where the R is has the value of the peak and zero in the other samples. At the end of the array of this input, a number concerning the individual

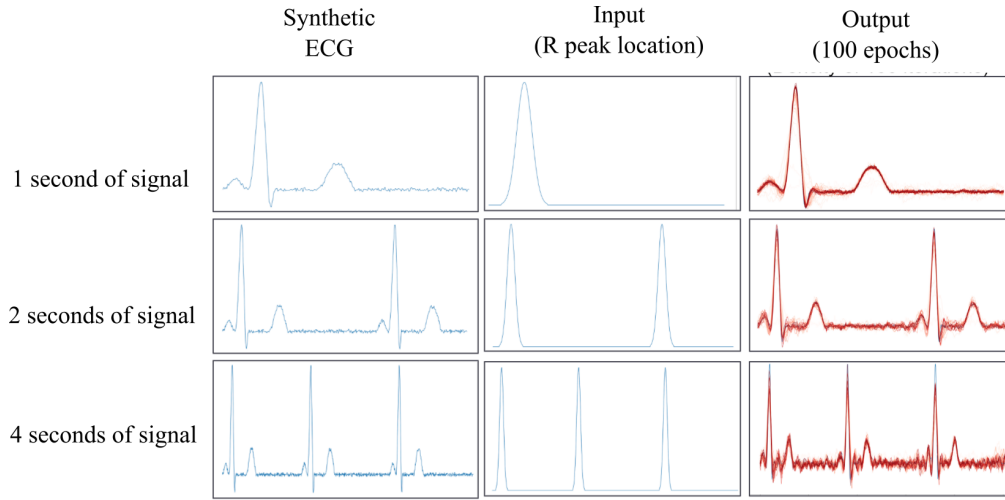


Figure 7.10: Reproduction of three examples of synthetic ECGs. The shades of orange are reproductions throughout the training epochs.

was concatenated. The first individual as a concatenated value of 0 while the second has 0.5. This is used for coding which "style" of ECG the network will generate. As seen, this network needs more fine-tuning to reproduce all signals in the test set.

As the ECG rhythms are modulated by other other signals, such as BVP and RESP, it could be possible to translate them using the R signal. While serving the R peaks as input and the desired signal to calculate the loss function, the synthesis of these signals is possible.

As for the implementation of these signals in the RNN, it is imperative to use an indicator of the phase of the cycle, as the samples need to have the information of the continuity of the values. Therefore the indication of the R peak could be done while using a sinus wave during the cycle duration, which will give the generator the an internal representation of the phase of the ECG dynamics. Another way would be to use the Syntactic Search in Time-Series (SSTS) connotation mechanism, and in this way, the output is a symbolic interpretation that will translate what is the stage of the cycle the signal is.

7.4.3 GAN Generation

The difference between the GAN generator and the previous is the insertion of a discriminator, capable of distinguishing between a fake generated sample, and a true sample from storage (Fig. 7.12). Both the generator and the discriminator must learn in turns. The generator should be pre-trained, and able to distinguish between noise and the true signal to help the training. The training should be in turns, wherein each epoch the discriminator weights are first locked, and its resultant loss will be used to train the generator, and after that, the discriminator can be trained in separate with the weights unlocked.

The loss function for the GAN generator should be the combination of the Mean Squared

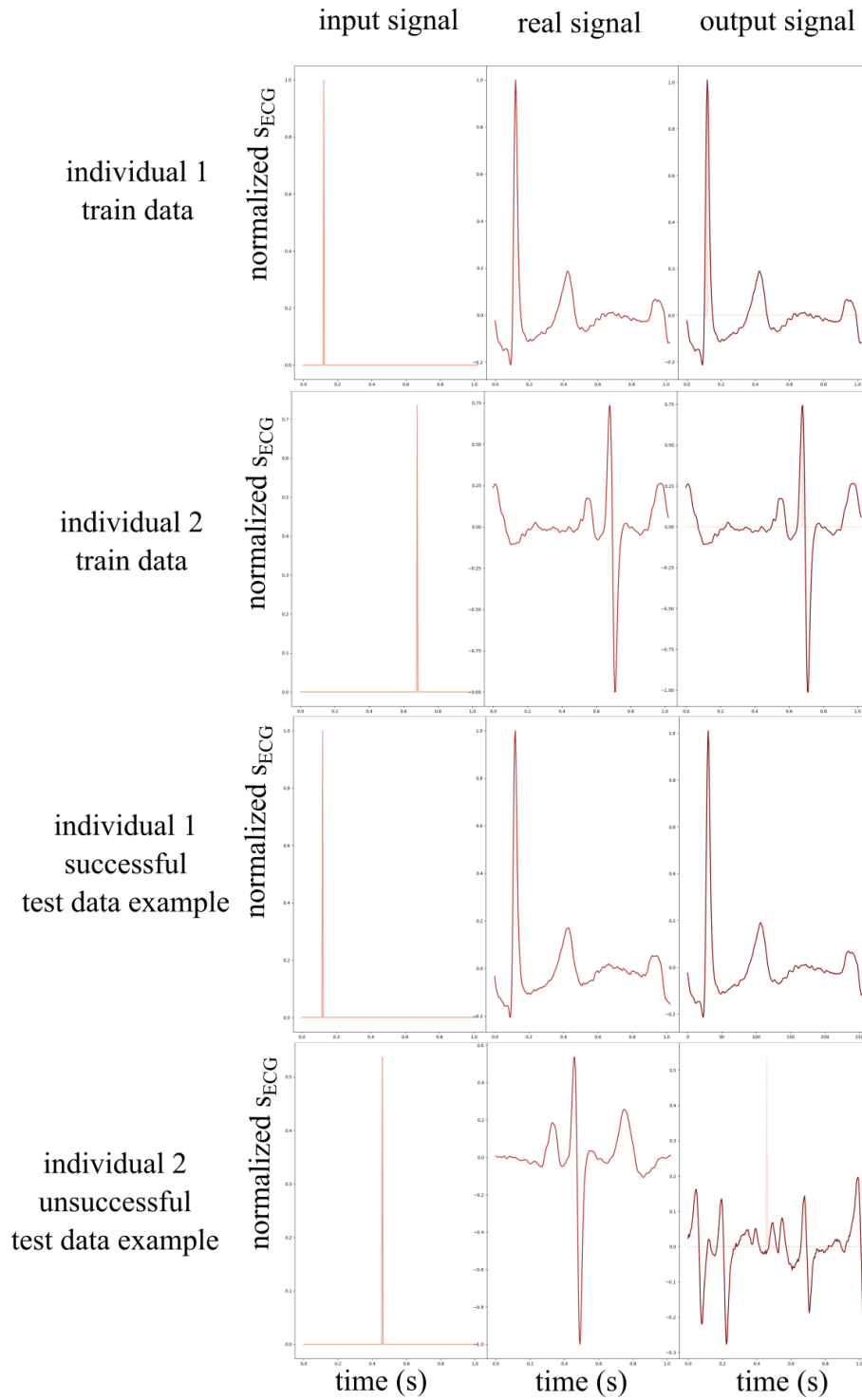


Figure 7.11: Reproduction of four examples of real ECGs.

Error (MSE) of the expected to be reconstructed with the inverse of the discriminator loss. As the discriminator gains lower values, the ability to reproduce is worst and vice-versa, and therefore adversary.

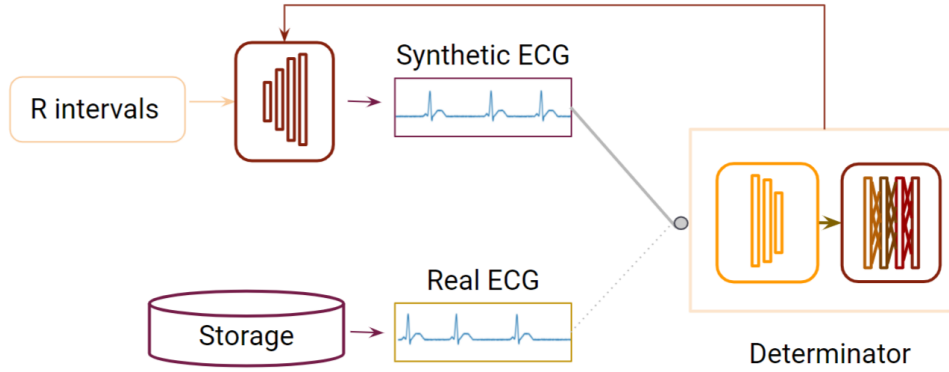


Figure 7.12: Diagram of a generator using a discriminator to create more realistic signals with an agnostic feature.

7.4.4 Mixture models

This architecture was produced to incorporate symptomatic events in a wearable ECG signal. Therefore to generate synthetic symptomatic data, this architecture was based on Fader Networks [178], which uses a discriminator to enforce certain attributes in the latent space of the encoder-decoder, by learning the agnostic representations of each class. In this case, latent representation is enforced to encode the desired attribute, which can then be fused with the generator to produce the desired class. This architecture, named ECG Generator of Representative Encoding of Style and Symptoms (EGRESS), depicted in Fig. 7.13, learns two major characteristics of the input ECG signals: the style and the content. All the modules present on the model are CNN and the structure is comprised of two autoencoders, two discriminators and one main generator.

The style of the signal is represented by the features that determine the acquisition device and conditions, whether it be wearable or a clinical setting. The content of the signal is the morphological characteristics of a specific heart condition. The hypothesis is that the synthesized signal should be a fusion primarily between these two components - it should consist of the "style" of wearable and the "content" of the asymptomatic signal. Therefore, the proposed architecture comprises a pair of encoder-decoder-discriminators that learn the respective aforementioned characteristics, as well as a generator that fuses the latent features learned from the encoder-decoder network.

The discriminators are first trained on a signal to determine the "style" or "content" of the signal. Therefore the encoder-decoders are trained with their respective discriminators, not only to reconstruct the original signal but also to maximize the confidence of the discriminator. Both autoencoders will extract the machine-computed features needed to reconstruct each one of the respective signals. One will be trained with the wearable and asymptomatic clinical data to learn the style features, while the other will train with symptomatic and asymptomatic ECG time-windows from the same data record to learn

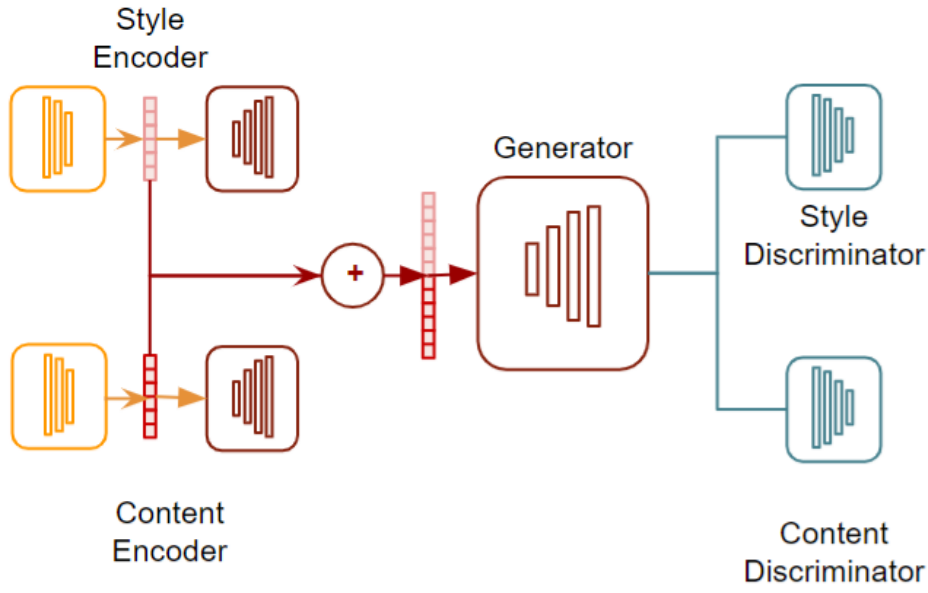


Figure 7.13: Diagram of the EGRESS generator that uses two encoders and two discriminators for a mix generation between style and content of two signals.

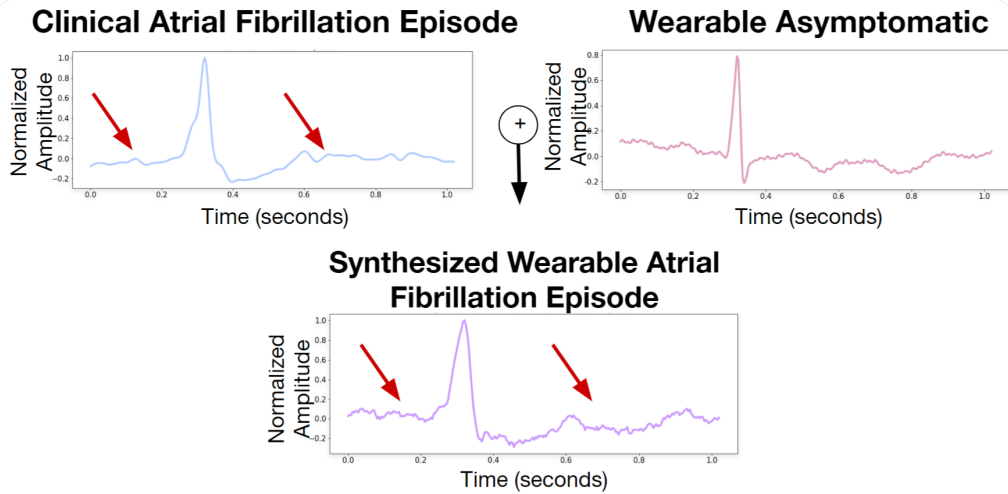


Figure 7.14: Example of a synthetic ECG combining a wearable asymptomatic ECG and a clinical grade symptomatic ECG.

the content features.

The resulting latent representations of the signal provided by the encoding part will be concatenated into a single vector and fed into the generator. This generator will learn how to produce a mixture of two signals based, not only on the error produced by comparing the input signals, but also comparing the error provided by the pre-trained binary classifiers, which we name as the discriminators in our architecture. As mentioned before, the style discriminator is trained to classify if the signal's origin is from a wearable device or not, and

the content discriminator is trained to classify between asymptomatic versus symptomatic clinical data.

The loss function of the generator is the weighted combination of the weights of each component of the network:

$$L_{gen} = w_{as} \cdot MSE(s) + w_{ac} \cdot MSE(c) + w_{ds} \cdot BCE(k_s) + w_{dc} \cdot BCE(k_c) \quad (7.1)$$

where w_{as} , w_{ac} are the weights for the MSE between the generated signal and the style input signal (s) and content signal (c) and w_{ds} , w_{dc} are the weights for the Binary Cross Entropy (BCE) for the classification of k_c and k_s , respectively. Fig. 7.14 depicts one example of the created windows using this approach.

Application

In this chapter, three different scenarios for the application of some of the aforementioned architectures are presented. These scenarios include the synthesis of three different biosignals, the use of Electrocardiogram (ECG) in biometric systems and, finally, the abnormality detection in the context of noise and arrhythmias of ECG signals.

In the first case, a model based on Gated Recurrent Units (GRU) learns and synthesizes biosignals, validated by the morphological equivalence of the original ones. This research leads to the creation of novel algorithms for data augmentation, signal reconstruction in heavily noisy data and source detection. It was tested for Respiration Signal (RESP), Electromyogram (EMG) and ECG signals. After the models were trained for each signal, they generated signals by forecasting a random value and re-feeding itself, similar to the morphological expression of the originals. During the learning process, after a set of iterations, the model starts to learn the basic morphological characteristics of the signal and later their cyclic characteristics. This synthesis mechanism has shown relevant results that inspire the characterization of signals from other physiological sources.

After comparing the error of prediction from a trained model with several ECG of different people, the opportunity of using it as a measurement for biometric studies was investigated.

With the increasing concerns over criminal acts, new solutions for identification and authentication arise. The biometric systems are based on a pattern recognition problem where the individual traits of a person are coded and compared. Biometric ECG systems fulfill the requirements but issues regarding intra-variability make real-life scenarios challenging. The output of this algorithm is used for both identification, i.e. the discrimination of the source of the signal, and authentication, i.e. the proof that the claimed subject is the source of the signal.

At last, the third application involves the exploration of the insights given by the

changes in the cardiovascular state. Two examples are the abnormalities given by the presence of noise that occurs during the acquisition, and the pathological events given by arrhythmias. Even though this process only learns the normal ECG signal, sacrificing the accuracy of detecting different types of events, it is fine-tuned to detect when changes occur in the signal. This approach could represent a major contribution to symptomatic screening, active learning.

All these studies will be presented in detail after a brief description of the used datasets.

8.1 Datasets

All data, except for the EMG used for synthesis, is publicly available and extensively used by research for training and validation of Machine Learning (ML) algorithms. The public datasets of Fantasia, ECG-ID, MIT-BIH can be accessed through Physionet, a biosignal database that was created under the auspices of the National Institutes of Health. The mission of these resources is to conduct and empower biomedical research and education. This platform also organizes conferences and annual challenges of unsolved problems in the clinical and basic sciences [179]. The CYBHi dataset can be obtained from the Zenodo research database stored in the CERN data cloud infrastructure to promote peer-reviewed open access research while curating the data uploads [180]. All the used datasets will be described in detail.

8.1.1 Fantasia

The Fantasia dataset was acquired from twenty young (21-34 years) and twenty elderly (68-81 years) while exposed to 120 min of continuous supine resting ECG and RESP (250 Hz sampling rate) while watching the Disney's movie Fantasia [159]. The signals collected by this database are mostly clean and free from symptoms, which reflects a good baseline for the collected biosignals. This dataset was used in all applications.

8.1.2 ECG-ID

ECG-ID consists of 310 recordings of 90 subjects aged from 13 to 75. While the original data ranges from 2 to 20 recordings per subject, only the first two were used in the ECG biometrics application. The purpose of this selection is to have the same number of samples for each subject and ensuring a balanced dataset. Each recording is 20 seconds long with a sampling frequency of 500 Hz [181].

8.1.3 MIT-BIH

The MIT-BIH Database has been available since 1999 in Physionet. The basic subset of this database, MIT-BIH Arrhythmia, contains ECG records from 47 subjects with 360Hz of sample frequency and 11-bit resolution, from Boston's Beth Israel Hospital. This

database contains 14 different types of arrhythmia, including Atrial Fibrillation (AFIB), Atrial Flutter (AFL), Ventricular Bigeminy (B), Paced Rhythm (P), Wolff-Parkinson-White Syndrome, or Pre-excitation (PREX), and Sinus Bradycardia (SBR) [114]. More subjects, with no significant arrhythmic episodes, were added to this dataset, 18 subjects in MIT-BIH normal signal rhythm (NSR), and 7 individuals in MIT-BIH Long-Term [114]. These collection of records were used for the abnormal episodes detection and biometry systems applications.

MIT-BIH Noise Stress Database is a part of this database which includes signals with noise artifacts such as baseline wander, muscle movement contamination and raw noise from electrode motion [114]. This data was used for the noise detection algorithm.

8.1.4 FMH Cycling EMG

This dataset, used for the signal synthesis, consists of EMG data that was acquired during an exercise performed in the faculty of human kinetics (FMH), University of Lisbon, Portugal, to measure muscular fatigue during cycling exercises. In this study, fourteen physically active, but non-expert in cycling, healthy males were exposed to a cycloergometer (Ergomedic 839E, Monark, Stockholm). These performed consistently, in a specific power setting, maintained until task failure while recording the EMG of the lower limbs (*rectus femoris* and *vastus medialis*). The acquisition system followed the directives of the International Society of Electrophysiology and Kinesiology (acquisition at a sampling frequency of 1000 Hz, filtering using a band-pass filter between 10 and 500 Hz and common mode rejection ratio of 110 dB) [182]. This dataset was used for biosignal synthesis.

8.1.5 CYBHi

The *Check Your Biosignals Here initiative* (CYBHi) dataset is a system that acquired ECG without the use of skin electrodes. The published work of Da Silva et al. (2014) [167] states that this work "devised a data acquisition framework and experimental setup, for large scale data collection from a large group of subjects through an easily repeatable and efficient procedure". Both sensors had hand-shaped support synchronized via syncPLUX synchronization kit, in which, the electrodes were placed on the fingers while the subjects were seated for 2 min in a resting position. The considered data for our work consists of the long-term sessions, where two acquisitions were made with 3 months difference, including 63 participants, 14 males and 49 females (18–24 years old) [167]. For the biometrics application example, the long-term sessions from the first acquisition will be named M1 while the second acquisition will be named M2.

8.2 Learning and Synthesis of Biosignals

Biosignal synthesis has been applied in Biomedical Engineering to mimic the morphological expression of these time-series. With the increase of computational power and the

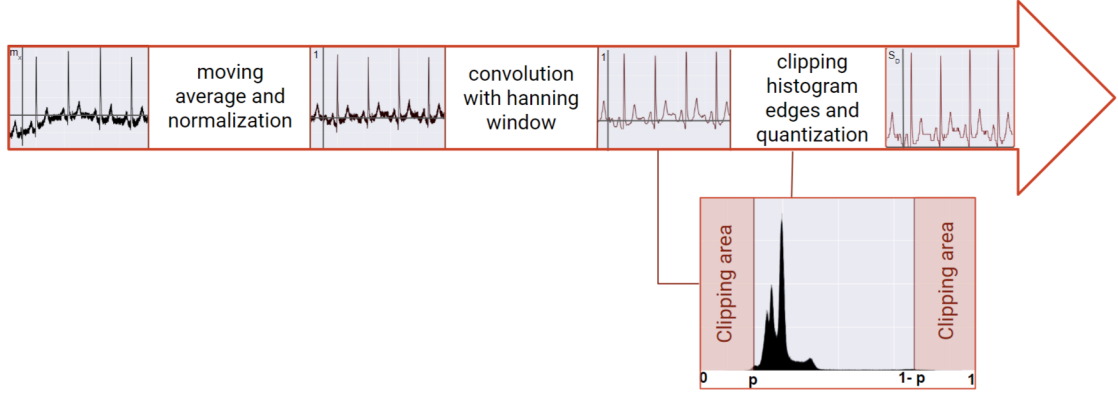


Figure 8.1: Signal pre-processing algorithm, based on character level language model.

capabilities of the Deep Neural Networks (DNN) to model complex structures, it is possible to create novel algorithms that can learn their morphological structure.

Since the state-of-the-art models only work with a specific type of biosignals, we hypothesize the creation of a DNN architecture could learn the inner workings of a time-series independently of the source or type of physiological signal presented. The resultant synthesis of these time-series with the learned models could lead to event detection algorithms capable of exploring disturbances of the morphology caused by artifacts or pathologies, reconstruction of physiological signals from heavily noisy data or the creation of systems of prevention and diagnostics of human dysfunctions by training other ML algorithms through data augmentation.

Before delineating the architecture, for the sake of establishing a common ground for interpretation, the etymology term morphology is *morph* - 'shape', 'form' and *logy* - 'study of', and together they are defined as the study of shapes or forms. In biology, this term is used in the study of form and structure of organisms, in geology the study of configuration and evolution of landforms, in linguistics the process of word-formation [183], and the term "Mathematical Morphology" is the mathematical concepts involved in analyzing the shape and form of objects [184]. For this work, we define biosignal morphology as the shape of its graphical representation, visualized by the human eye. This definition is augmented by how the average mind perceives the periodicity, amplitude, structure, disruptions, and clearness in the form of the signal. The similarities and differences between the morphology of the biosignals will also be analyzed empirically.

As depicted in chapter 2, different biosignal groups are morphologically different from each other. While observing the figures displayed in Fig. 8.2 it is possible to identify 8.2 (a) and 8.2 (b) as being both ECG, and therefore, clearly closer morphologically than 8.2 (c) or 8.2 (d). Differences between families a clear morphological divergence are visible, because of the different physiological phenomena. In Fig. 8.6 the EMG wave may be interpreted as a signal with a high frequency, in which the signal has a periodic change of amplitudes. As for the RESP signal, the wave is slow that has small changes in its

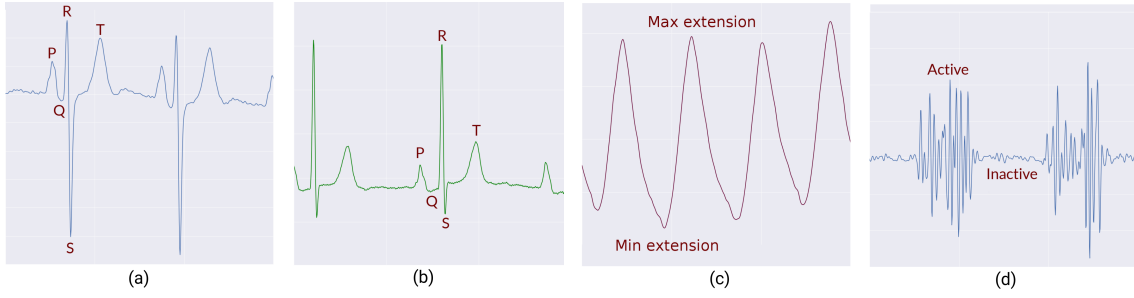


Figure 8.2: Four different acquired biosignals: In (a) and (b) two ECG signals from two different subjects; in (c) an example of an EMG signal; and, in (d) an example of a RESP signal.

frequency when an individual is breathing normally but may gain higher frequencies when under spasmodic events.

To reproduce these complex and different behaviors, a DNN is used as the automatic learning mechanism.

8.2.1 Methods

The chosen dataset was based on three principles: free from noise; acquired from individuals without pathologies; and, the signal morphology must be directly interpreted by a human without any special expertise. Consequently, the Fantasia dataset, for both ECG, RESP, and the EMG signals, acquired from the cycling exercise, are the a good fit [159, 179].

As stated in Fig.8.1, for the signal to have a homogeneous amplitude through all signal, each sample was normalized using a moving absolute maximum window and subtracted by a moving average window (eq. 5.4) and smoothed with a moving average Hanning window with the size of 10 samples. Before the quantization process, the extremities of the signal were removed, mitigating the influence made by artifacts. This was achieved by clipping the edges of the amplitude histogram of the ECG within a parameterized value of confidence, depending on the presence of artifacts that corrupted the signal with high amplitudes of the signal. These types of artifacts disrupted the normalization and quantization processes. The value of this parameter is typically 0.5%. In the end, the signal was segmented into D time windows (TWs) with dimension W , resulting in two matrices X and Y with the same dimensions of $D \times W$, where Y is the same signal as X with one sample delay.

The DNN¹ architecture is depicted in Fig. 8.3, in which a sequential approach is proposed. The scalar x_n is transformed into a vector \hat{x}_n by selecting the corresponding column of the embedded matrix - E . The result of the three GRU layers is vector \hat{o}_n that will be the input of a regression node and a softmax function.

¹The used library to process the Artificial Neural Networks (ANN) algorithm was *Theano* [185], a Python library that allows to define, optimize and evaluate mathematical expressions involving multi-dimensional arrays efficiently. It has been one of the most used CPU and GPU mathematical compilers, and has been used to produce many state-of-the-art machine learning models since 1998 [185].

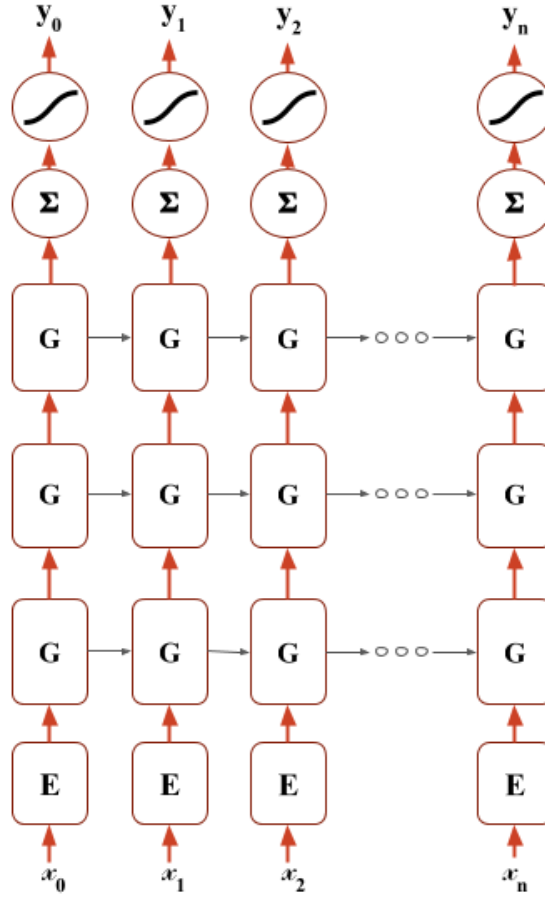


Figure 8.3: Sequential DNN model. This model comprises one embedded layer - E , three GRU layers, one regression node and one *Softmax* function. The variables above the arrows represent the output and the input of the previous and the following node, respectively.

8.2.1.1 Training

The model was trained using RMSProp and the loss function was the cross-entropy loss (eq. 3.17). Due to computational limitations, the data was divided into a fixed number of TWs batches - B_D . Each batch was submitted as input to train the model. After a fixed number of epochs, the model accepts the next batch of data and repeats this process until all the TWs of each signal is seen by the model.

8.2.1.2 Signal Synthesizer

After training the model, the synthesis of the signal was performed by re-feeding the input of the model with the last prediction. The selected value will a semi-random choice based on the probability density function given by the output of the softmax function. The generated signals were based on a model that was trained with the referred three distinct signal types for each individual, totaling 54 distinct models.

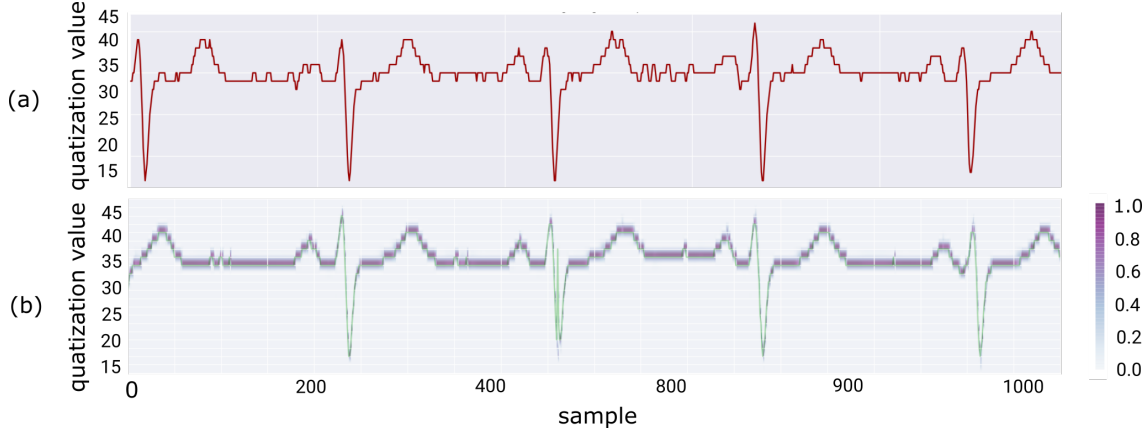


Figure 8.4: ECG prediction for subject 3 of fantasia dataset. (a) depicts one TW of the pre-processed signal of the original dataset. (b) The green signal is the generated signal with the DNN model for this subject and in purple the probability of each class - k - given by the output of the previous sample.

8.2.1.3 Model Evaluation

For the evaluation of the models and to guarantee that the signals were independently modeled, the Mean Squared Error (MSE) was calculated for each batch of data for each signal and model:

$$\text{Error}(B) = \frac{1}{N \cdot B_D} \sum_{b=0}^{B-1} \sum_{n=0}^{N-1} (\hat{y}_n^b - y_n^b)^2 \quad (8.1)$$

This measure is closer to the morphology validation than the cross-entropy error. The premise is that the prediction error given by the model that was trained with a specific signal should be higher when fed with signals that were acquired from different sources.

Each signal was pre-processed and separated into a training and a testing set: 128 random TW of the first 33% of the signal were used for training; 66% of the signals were used for the test. The test windows had a size of 512 samples and the number of windows dependent on the size of the signal.

The mean and standard deviation were calculated for all windows, for each signal and trained model.

8.2.2 Results

The created DNN model was tested in three types of biological signals, as stated before: ECG (Figs. 8.4 and 8.7); EMG (8.6); and, RESP (8.5). This section also covers the information related to the learning stages of the algorithm adopting the ECG as input to guide the explanation.

8.2.2.1 Electrocardiogram Generator

The first experiences were made in an ECG of an individual with a high S_D (number of quantization steps) value, turning it hard to compute due to the number of possible values the network has to process and select. Therefore, it was established that in the first stages the algorithm would learn with low resolutions. One representative example of these experiments is depicted in Fig. 8.4. The graph in Fig. 8.4 (a) is a segment of the ECG that belongs to the subject 3 of the *Fantasia* dataset [159] and the green graph is the synthesized version. This graphic is the result of a model that was trained with a $S_D = 32$ and $H_D = 32$. To present the most important features of the signals in each TW while training, the chosen value of W was 256, reflecting the sampling frequency (250Hz) and the period of a normal ECG (60bpm). After some trial-and-error, it was understood that this parameter was important for the model to learn the morphology of the signal.

All the ECG characteristics are visible both in the original and synthesized signals. Not only the model learned the frequency and the principal characteristics of the compound wave, but also that it has a baseline at $k \simeq 40$ and the values of the local minima and maxima. It is also observed that at the beginning of the synthesized signal, that a sample in the QRS complex was not predicted successfully by the generator, but the morphology regains the expected shape but after that, the signal is morphologically close to the original signal.

One further aspect of this ECG synthesis is the fact that the network also learns the individual morphology of the person's ECG. In Fig. 8.7 the synthesized ECG produced by the model trained with subject 7 of the *Fantasia* dataset is different from the one created from the subject 3, depicted in Fig. 8.4.

8.2.2.2 Respiration Signal Generator

As depicted in Fig. 8.5 the model learned the patterns, the amplitude and even the small differences in breathing frequency throughout time. The two small discontinuities in the start and end of the signal are the only aspects that illustrate the model's imperfection. Since this signal is slower than the previous ones, the W size had to be higher for the model to learn the respiratory signals but needed fewer epochs and TWs to give satisfactory results.

8.2.2.3 Electromyogram Generator

Fig. 8.6 (a) reflects the EMG signal of the *vistus lateralis* muscle while cycling with the pedals attached to the shoes. This EMG may be interpreted as a signal with a high frequency, in which the signal has a periodic change of amplitudes. The active phase is originated by the muscular contraction while pulling and pushing the pedal.

To get a full cycle in the input, the model required a higher W value since the sampling rate was 1000 Hz. The selected dimensions where $H_D = 64$ and the $S_D = 128$ that even

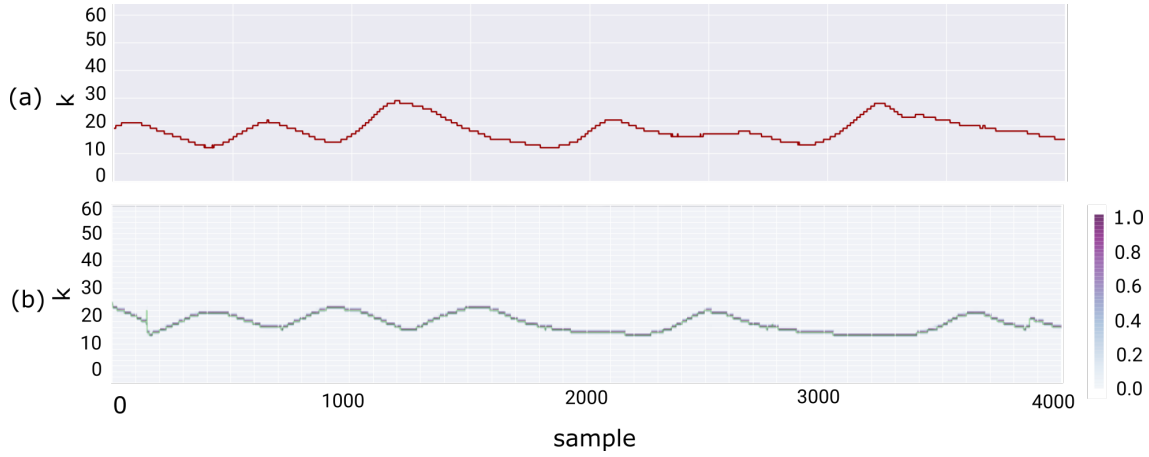


Figure 8.5: RESP prediction for subject 3 of fantasia dataset. (a) depicts one TW of the pre-processed signal of the original dataset. (b) The green signal is the generated signal with the DNN model for this subject and in purple the probability of each class - k - given by the output of the previous sample.

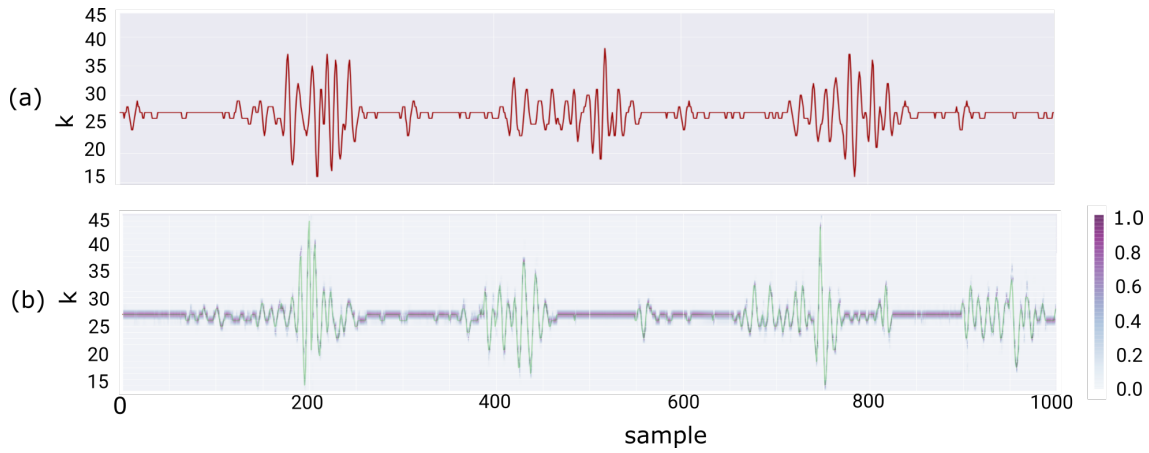


Figure 8.6: ECG prediction for subject 3 of fantasia dataset. (a) depicts one TW of the pre-processed signal of the original dataset. (b) The green signal is the generated signal with the DNN model for this subject and in purple the probability of each class - k - given by the output of the previous sample.

though the epoch time (t) was higher than the previous signal (ECG), the data required to learn was smaller. In the predicted signal (Fig. 8.6 (b)), the cycles are visible. In the original signal (Fig. 8.6 (a)) a rhythmic noise caused by a small amplitude ECG is visible in the rest areas of the signal. Even though the model did not learn the frequency accurately, its morphology and the reproduction of this artifact is present.

8.2.2.4 Model Evolution

The model exemplified in Fig. 8.7 is the synthesis of the individual number 10 of the *Fatasia* dataset. While the model was being trained, several graphical representations of how the model was learning were made. Therefore it is depicted as six generated signals

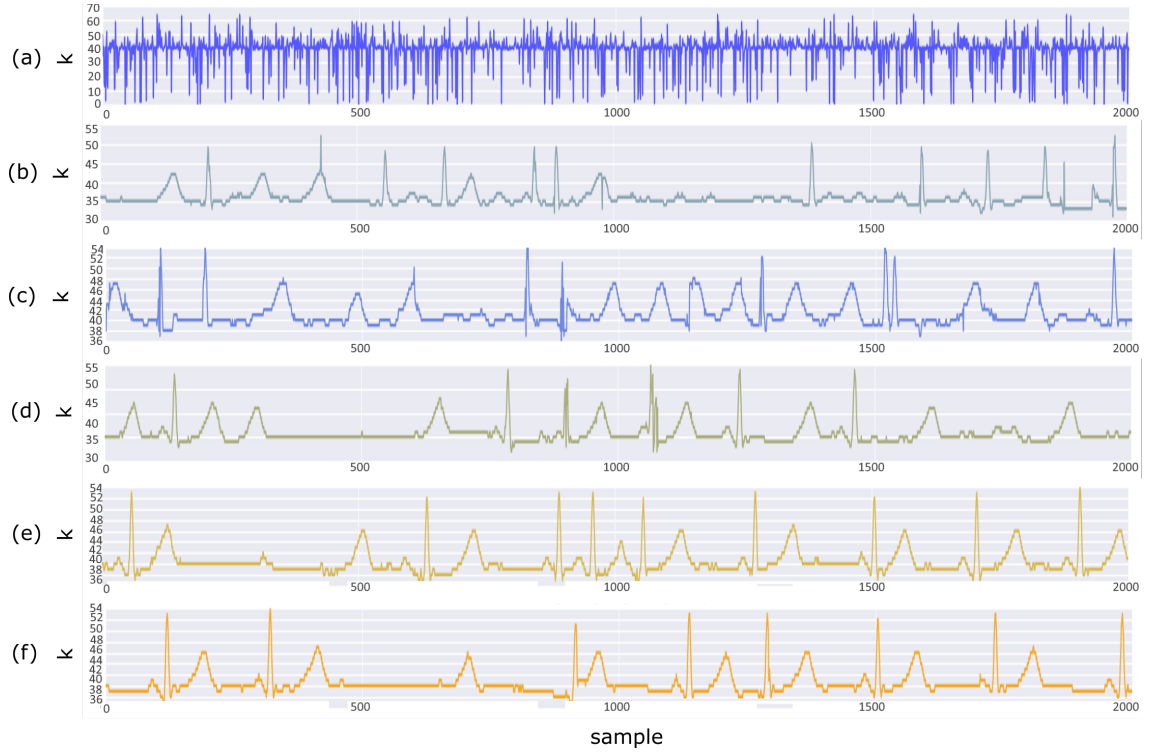


Figure 8.7: History of the ECG DNN model with $S_D = 64$ and $H = 64$ for subject 10 of Fantasia dataset. Each graphic depicts the predicted signal for a model that was trained after a specific number of epochs during a specific batch of data: (a) 0 epochs for batch with TWs from #0 to #31; (b) 60 epochs for batch with TWs from #0 to #31; (c) 90 epochs for batch with TWs from #0 to #31; (d) 90 epochs for batch with TWs from #32 to #63; (e) 90 epochs for batch with TWs from #192 to #233; and, (f) 90 epochs for batch with TWs from #1120 to #1151.

for different numbers of trained batches and epochs. For example, in the first graphic (Fig. 8.7 (a)), the first batch of 32 TWs is a result of the model before it was trained. As for the second graphic (Fig. 8.7 (b)), it is the result of the prediction of the model trained with the same batch but after 60 epochs.

Looking again to the first stage, the model parameters were initiated with random values, therefore, the plot presented is a sequence of values with a mean value and a standard deviation. After a few epochs, the model learned a few characteristics of the signal, as seen after 60 epochs where some rudimentary forms of the ECG wave, such as T or P waves, and some R spikes are presented. Although there is no beat frequency, the signal returns to the base value, in this case, approximately 40. After 30 epochs the waves become more defined, and the R peaks are resembling more the final version. While training the second batch (Fig. 8.7 (c)), the model introduces at least one slow wave before and after the QRS complex, and, even though it misses at times a P or a T between two R, it doesn't repeat these waves. Finally, after 1149 TWs, even though there is a latency in the first samples, it is possible to see periodicity. The definition of the Paced Rhythm (P) Q, R, S and T waves and its sequence also reflects the original ECG characteristics

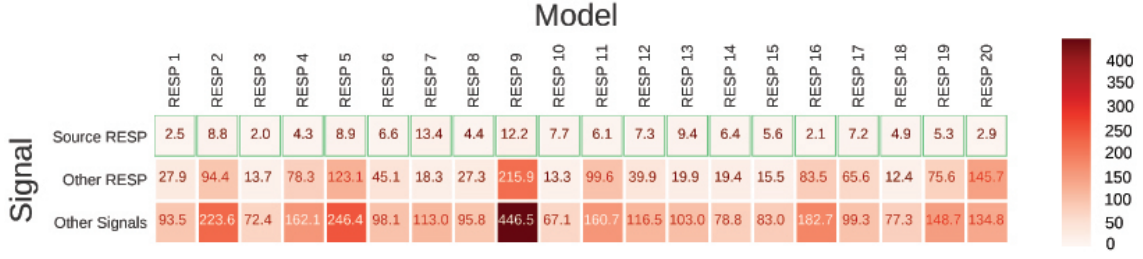


Figure 8.8: RESP test results for error average. The green square represents the lowest value for the correspondent model. "Other RESP" represents the error of predicting respiratory signals other than the training source. "Other Signals" represents the error of prediction of the EMG and ECG signals.

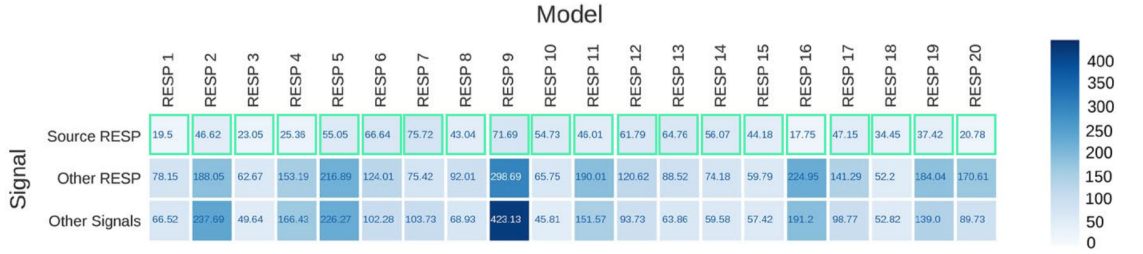


Figure 8.9: RESP error standard deviation. The green square represents the lowest value for the correspondent model."Other RESP" represents the error of predicting respiratory signals other than the training source. "Other Signals" represents the error of prediction of the EMG and ECG signals.

(Fig.8.4 (a)). It was visible that after a certain number of batches, the model overfits data and unlearns.

8.2.2.5 Model Evaluation

For the model evaluation, the prediction error for each type of signal is presented in Figs. 8.8, 8.9, 8.11, 8.10, 8.12, and 8.13. The average and the standard deviation values were calculated with the same window size ($W = 512$) for each signal and model. None of the widows of the testing group were fed to any model while training. The number of TW depended on the signal size, for RESP and ECG the $B_D = 3584$ for each and for EMG was $B_D = 612$. The green squares represent the lowest averages for each signal each column. The first-row source signal is relative to the signal that trained the correspondent model, as for the second row refers to all the other signals of the same type while the third is linked to the rest of the signals.

8.3 Biometric Systems

The published work of Jain, Flynn, and Ross (2007) [186] defines biometrics as "the science of establishing the identity of an individual based on the physical, chemical or behavioral

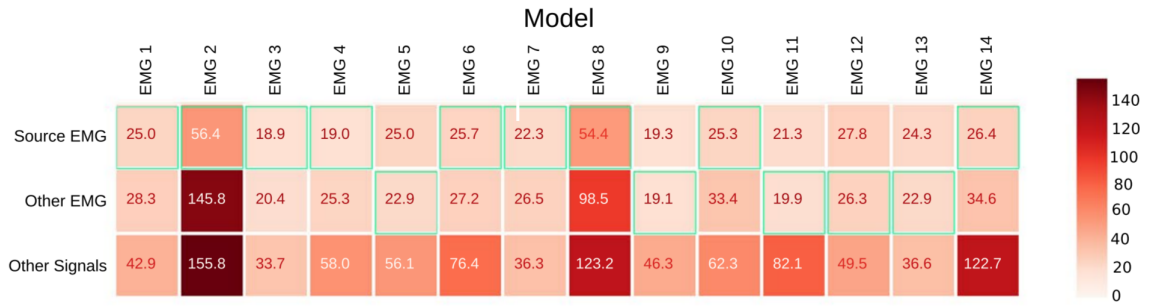


Figure 8.10: EMG error average. The green square represents the lowest value for the correspondent model. "Other EMG" represents the error of predicting EMG signals other than the training source. "Other Signals" represents the error of prediction of the RESP and ECG signals.

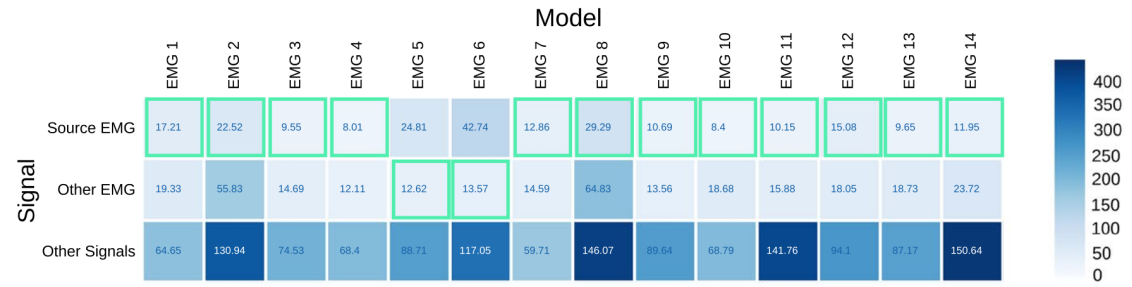


Figure 8.11: EMG error standard deviation. The green square represents the lowest value for the correspondent model. "Other EMG" represents the error of predicting EMG signals other than the training source. "Other Signals" represents the error of prediction of the RESP and ECG signals.

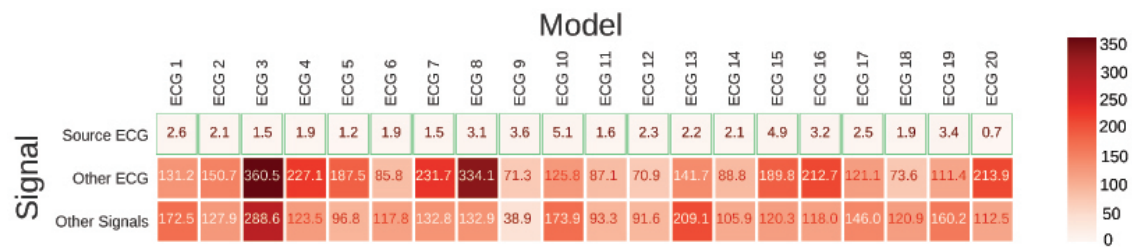


Figure 8.12: ECG error average. The green square represents the lowest value for the correspondent model. "Other ECG" represents the error of predicting ECG signals other than the training source. "Other Signals" represents the error of prediction of the RESP and EMG signals.

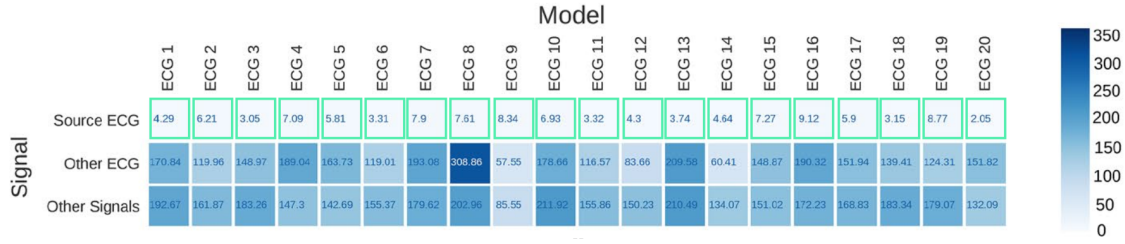


Figure 8.13: ECG error standard deviation. The green square represents the lowest value for the correspondent model. "Other ECG" represents the error of predicting ECG signals other than the training source. "Other Signals" represents the error of prediction of the RESP and EMG signals.

attributes of the person" [186]. In sum, this pattern recognition problem compares the individual physical, physiological, and behavioral coded traits of a specific person with the records within a database which contains the same biometrical features.

There are essentially two different biometric tasks: identification and verification. Identification is a 1:N comparison, with the primary goal of finding which person registered in the database is the source of the provided biometric data sample. In the verification mode, also known as authentication, a 1:1 comparison is made, where the system will decide whether the provided data belongs to the claimed person or not. Some authors also claim the screening modality, consisting of a 1:N comparison aimed to determine if the user is not in the database, usually for security checks [187–189].

Typically, a biometric system comprises [188]:

- Acquisition module - the acquisition of the biometric data;
- Feature extraction module - the processing of the data to find relevant discriminatory traits;
- Match module - compares the features of the sample with the database, requiring a decision algorithm to verify or reject the given identity vs. the given data;
- Database module - the container of all the biometric collected data.

The main advantages of these systems are the absence of previous knowledge such as passwords, and biometric attributes generally cannot be lost, transferred or stolen, even though some could be reproduced, the forgery is difficult and require the presence of the individual. But for these systems to be considered, they should follow a set of requirements, such as [168, 188]:

- Universality - everyone should possess the biometric modality;
- Uniqueness - level of difference between people;
- Permanence - invariance or stability over time;

- Measurability - difficulty of acquisition;
- Performance - the reliability of the method;
- Acceptability - subjects acceptance;
- Circumvention - sensitivity to fraud

Since the application of the elementary fingerprint recognition in 1883, research has been searching for different and reliable types of biometric data [190].

Nowadays fingerprint, face, hand, veins, voice, iris, and other biological and behavioral features are used to identify subjects [191, 192]. For example, in the context of the physiological signals, *i.e.* electric signals from the electrochemical changes in nerve cells, muscles, and gland cells, may be acquired from surface electrodes in contact with the skin. The ECG, from the heart, the EMG, from the muscles activity, and the Electrodermal Activity (EDA), that tracks the sweat level changes, representing the sympathetic nervous system activity, and the Electroencephalogram (EEG) [193], from brain activity, are some of the data used in biosignal biometrics [188, 194]. The focus of this section will be ECG biometric systems.

8.3.1 ECG Biometrics

Given its morphological signature and advances in sensing devices, the ECG emerged in the last decade as a biometric modality with the promise of robustness against circumvention attacks, and the ability to continuous pervasive acquisition scenarios. The main difficulties that involve the non-invasive ECG measurements are the intra-subject variability, artifacts and noise [194, 195] making the computation of the features arduous, particularly when the ECG signal is contaminated with noise in the characteristic points of the waveform [196]. Merone et al. (2017) [158] compiled several sources of artifacts and noise, such as electrode material, sensor locations, power-line interference, movement artifacts and instrumentation of the devices. Other variables that change the waves may reside in the health status of the patient, the heart-rate variability, physical exercise, affective status, and drugs, but Wübbeler et al. (2007) [197] states that the ECG remains stable over the years, the sufficient to allow recognition with an error rate of 3% [194, 195, 197].

In a typical ECG biometric system the data is submitted to a feature extraction module. After extracting features in the enrollment stage and a classifier is trained, where the conjunction of features corresponds to a subject. There are two main types of ECG biometric systems: fiducial-based, in which a representation of the ECG cycle is made, named template, in which the relevant points are calculated; the non-fiducial-based, which considers a ECG signal as a whole, calculates the features on this premise. Some studies use the combination of both methods, named partially-fiducial, to achieve better results [162, 194, 198].

In the fiducial-based methods, the selection of features may be temporal features, temporal intervals among reference points, such as wave peaks or boundaries, amplitude features, the relative height of the different waves to the R peak, or the combination of both. Algorithms based on non-fiducial features usually include segmentation of the signal and R peak detection for characterization of the individual [194].

Deep Learning has recently gained relevance in ECG biometrics field, several studies comprise one or more Convolution Neural Networks (CNN) modules that both processes, by learning features, and classifies by learning the relations between the learned features and the users. The input may vary, studies such as Labati et al. (2018) [199] uses the raw signal in a fiducial system but Zhang, Zhou, and Zeng (2017) [108] describes a non-fiducial system that feeds the network with Temporal frequency spectrograms obtained by discrete wavelet-transform [108, 199].

8.3.2 Objective for Biometrics based on ECG

The main objective is to improve current results on both identification and authentication in two different scenarios. The first scenario involves exploring a feature extraction mechanism by using spectrograms and the second does not use a human-crafted feature extraction module.

In the first scenario, two 2D CNN architectures will be explored while trained and validated with two databases: Fantasia and ECG-ID.

In the second scenario two architectures will be tested:

- The first architecture, depicted in Fig. 8.17 (a) is a direct application of the GRU-based architecture used for signal synthesis. The resultant biometric system will be non-fiducial, that has as a base hypothesis that if a model that is trained with signal A and fed with signal B from another individual, the prediction error will be higher than if fed with a signal with the same individual as A;
- The second system Fig. 8.17 (b), combines fiducial and non-fiducial elements and exploits the automatic feature learning of the CNN providing a score, where its higher values reflect a lower probability of the signal being of the corresponding subject.

In the end, the classification will be given by the relative score threshold classifier, which is a simple classifier that will compare the output scores and find the most probable source for the given sample. This scenario will be tested in these three databases: Fantasia, MIT-BIH and CYBHi [114, 158, 159, 167]. To test the robustness of these systems Gaussian noise is added in the Fantasia database.

8.3.3 Spectrogram Analysis Approach

This approach comprises 3 steps: preprocessing, spectrogram generation and subject identification through a CNN.

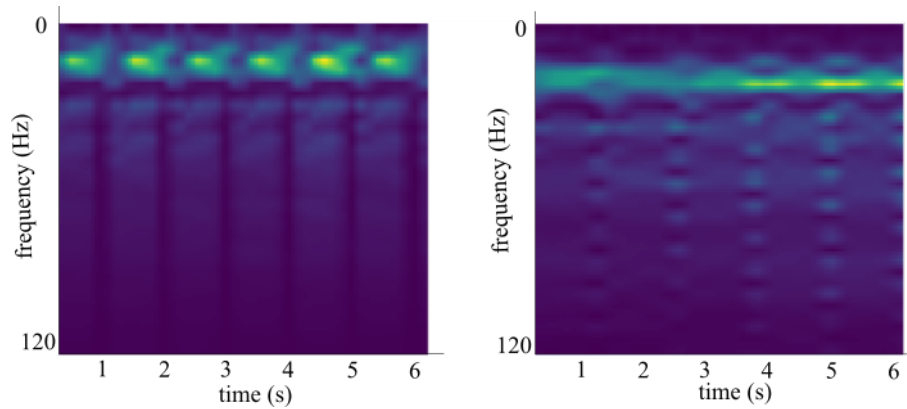


Figure 8.14: Two examples of spectrograms for each used database: Fantasia (left) and ECG-ID (right).

The signal preprocessing included the sequential steps of a "min-max" normalization (5.3), smoothing and average removal. After this the Fast Fourier Transform (FFT) spectrogram is created, which is the extraction of the magnitudes of the corresponding frequencies through time. The graphical representation has in the x-axis the time, in the y-axis the frequency and the energy in the z-axis, the dimension that was converted to color as depicted in Fig. 8.14.

8.3.3.1 Architectures

The 2D transformation of an ECG signal is used as input to two different networks: a simple CNN (Fig. 8.15) and a DenseNet (Fig. 8.16) based architecture [173]. In this type of architecture, the inputs of a layer contained in a block are a concatenation of all the previous feature maps. Besides allowing more efficient use of the network parameters, this type of architecture has achieved state of the art results on popular image recognition datasets [173]. After testing and optimizing the algorithm, the final developed DenseNet has 19 layers, including 3 dense blocks with a growth rate of 10, starting with 20 kernels on the first convolutional layer.

8.3.3.2 Results

Both architectures were trained using Adam optimizer ². Since the first architecture was first used to optimize the preprocessing stage, only a subset of the training data was used.

The signals were trained on 67% of the data and validated on the remaining 33%. For the Fantasia database, spectrograms were randomly chosen throughout the entire length of every signal, as the recording time is much longer when compared to ECG-ID, of which all the generated spectrograms were used.

²Used the Keras package [200] using TensorFlow [201] as backend with the computational power of a Nvidia Geforce GTX 1080 Ti Graphics Processing Unit (GPU).

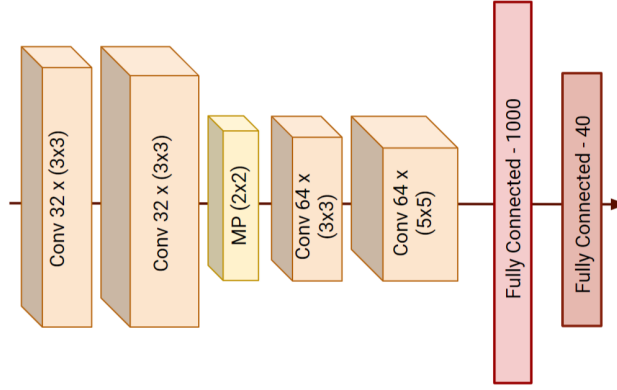


Figure 8.15: CNN Architecture for Fantasia database. It is composed by 4 Convolutional layers ("Conv") with a Max Pool ("MP") in the middle. The Classification layers in the end are composed of a Fully Connected network with 1000 units in the first layer and 40 in the last layer, corresponding to the number of individuals.

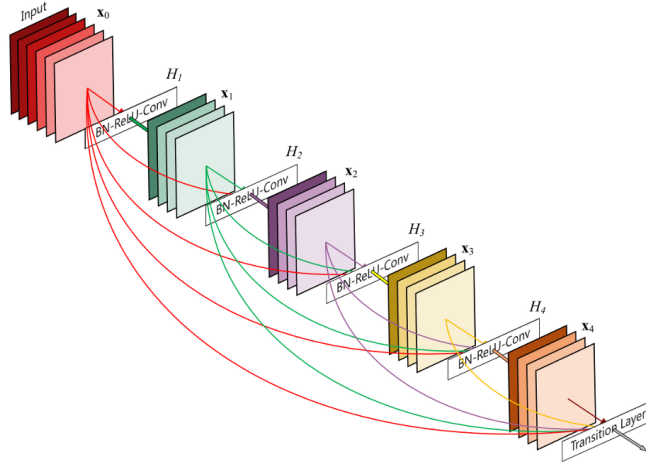


Figure 8.16: Example of a 5 layered dense block with a growth rate of 4. From [173].

Since both databases have a significant difference in the recording duration, the need for a minimum amount of training and validation data required a change of the used parameters for spectrogram generation in both databases.

For the Fantasia database, the signals are segmented into batches with a length of 1536 samples (approximate duration of 6 seconds) and the FFT window has a size of 256 samples with an overlap of 87.5%. For the ECG-ID database, the segments consist of 2048 samples (approximately 4 seconds) and a FFT window size of 512, with an overlap of 93.75%. The decrease of the segment duration reflects the need to increase the number of generated windows for training and validation, due to the shorter amount of samples. In both databases, all the segments are convoluted with a Tukey [202] window with a α parameter equal to 0.5.

Once the spectrograms are generated, the frequencies above 120 Hz are removed due

Table 8.1: Classification performance(%). "DB", "SC", "DN", "Fant." and "ID" stand, respectively, for Database, Small CNN, DenseNet, Fantasia and ECG-ID.

Model (DB)	Acc (%)	Sens (%)	Spec (%)
SC (Fant.)	99.16	99.42	99.98
DN (Fant.)	99.16	99.78	99.99
SC (ID)	94.23	91.87	99.94
DN (ID)	96.88	91.87	99.96

to the lack of information provided, the resulting matrix is resized to (80×80) and scaled by subtracting the mean and dividing by the standard deviation. Examples of the spectrograms are shown in Figure 8.14.

The results presented in Table 8.2 suggest that the combination of DNN and spectrograms can be used effectively for human identification, only needing a recording of 6 seconds to be able to accurately classify a subject on a universe of 40. The performance of these algorithms reaches state of the art for both Fantasia and ECG-ID databases, but even though the values of sensitivity and specificity presented in Table 8.1 are reasonable, there is a margin for improvement.

Our models are not as accurate on the ECG-ID database (Table 8.3) due to the short recording time, only allowing the generation of 366 spectrograms per subject, even with a considerable overlap percentage (93.75%), which may create redundancy of information while training. The accuracy values did not achieve the highest values compared to the related works, nonetheless, they are close enough to be considered state-of-the-art, having in mind that only a subset of the database was used and in the analyzed bibliography the diversity in subsets turns comparison difficult. Past experiments on ECG-ID database (Table 8.3) such as Lugovaya Lugovaya (2005)[181], Tan and Perkowski (2016)[203] and Page, Kulkarni, and Mohsenin (2015)[204] the dimension of the training and validation subsets diverged considerably, in the case of Salloum and Kuo (2017)[205] each subject was trained and tested with one session, not testing the effect on different time recordings and the inputs were not consistent, as the algorithm was fed with arrays containing from 3 to 9 heartbeats. This method was consistent, as it used the 2 sessions per subject and 4-second segments. As there are no widely followed standard procedures for ECG biometrics tasks the works cannot be compared equitably.

8.3.4 Through Signal Morphology

As stated before two architectures were compared and explored using the same classification method, the relative score threshold classifier. The setup, which is depicted in Fig. 8.17, comprises the Temporal Convolutional Neural Network (TCNN) network which follows a standard machine-created feature extraction and classification procedure while each Recurrent Neural Networks (RNN) network is trained for each individual creating an

Table 8.2: Accuracy comparison for Fantasia Database

Work	Accuracy(%)
Tantawi et al.(2013)[161]	95.89
Zhang, Zhou, and Zeng(2017)[206]	98
Zhang, Zhou, and Zeng(2017)[108]	97.2
Ours (Small CNN)	99.42
Ours (DenseNet)	99.79

Table 8.3: Accuracy comparison for ECG-ID Database.

Work	Accuracy(%)
Lugovaya(2005)[181]	96
Tan and Perkowski(2016)[203]	98.79
Page, Kulkarni, and Mohsenin(2015)[204]	99.96
Salloum and Kuo(2017)[205]	100
Ours (Small CNN)	94.23
Ours (DenseNet)	96.88

internal representation of the wave. The distance between the predicted wave and one given wave gives an error value for each model. When normalized, this error gives a score to which model it belongs to.

To train and test these models some considerations are made regarding the corruption of some segments. The heavily corrupted windows were removed using two different methods that assume that most of the signal is clean:

- The first extracts the windows that are between intervals around a percentage of the median of the standard deviation value and the median of the mean value of each window, depending on the database noise factor;
- The second uses the algorithm developed by Rodrigues, Belo, and Gamboa (2017) [195] relying on the standard deviation feature. This method was used for the CYBHi database where the first method was inefficient due to the low signal-to-noise ratio.

8.3.5 Recurrent Neural Network Architecture

This first architecture the same as applied in the synthesis, presented in detail in section 8.2, which is a RNN comprised by an embedded matrix E , three sequential GRUs G , a single-layer network and a softmax nodes. The data is transformed through the same sequential approach where the signal is first preprocessed and then fed to the RNN (Figure 8.3) where is trained for each different signal with the parameters of $H_D = 256$ and $S_D = 256$.

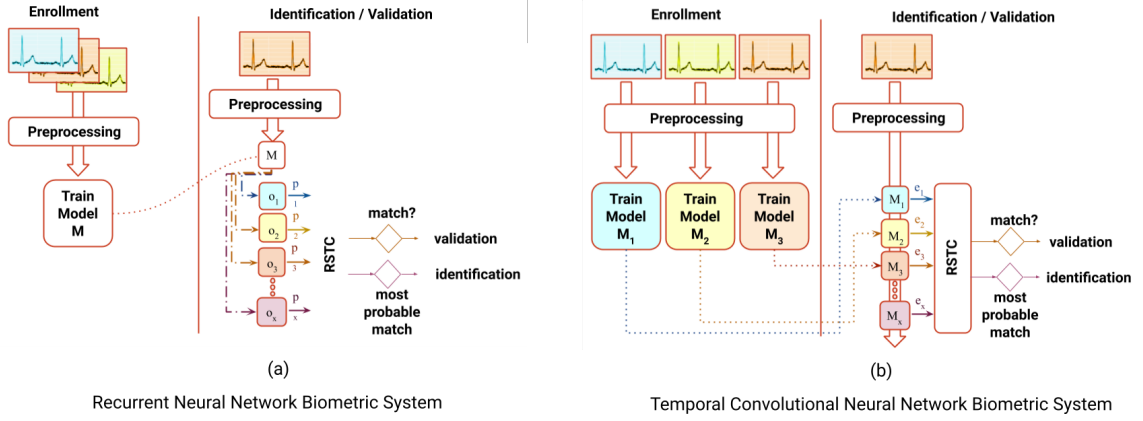


Figure 8.17: Proposed biometric systems. In Fig. (a) the trained model extracts features and has an output, that output is used to calculate the probabilities for each class (p), as for Fig. (b) the error of prediction (e) of each model is used. Relative score threshold classifier is the classifier that validates or identifies which person does the input sample belong.

Since this architecture can learn and synthesize an ECG signal and proved by the fact that the prediction error of a model is lower when the input signal is the one that trained the model concerning the other models, even though it was never trained with the tested sample, the hypothesis is expanded to the individual source of the signal. The hypothesis is that the network is also capable of learn the individual intricacies of each person's ECG. Following this reasoning, after the set of errors is normalized, it can be concluded that lower values are correlated with higher similarity between the input and the ECG which trained the model. The produced similarity score ($S(p, i, w)$) is given by the following equation:

$$S(p, i, w) = \frac{o_p}{\max(o(i, w))} \quad (8.2)$$

, where p is the predictor, i the individual, w is the time-window, and $o(i, w)$ is the output for all the models for that individual and time-window.

8.3.6 Temporal Convolutional Neural Network

The second proposed method is a two-stream TCNN [207], which uses 1-dimensional convolutional layers with dilated convolutions to learn temporal patterns. It combines predictions from two different inputs: ECG non-fiducial window segments and a sub-segment of the same window containing only one full cycle centered in the QRS complex.

This architecture is depicted in Figure 8.18, where each network is composed of several convolutional layers with 24 kernels of size 4, followed by batch normalization and a Rectified Linear Unit (ReLU) activation function. At the end of each network, two fully connected layers, the first with 256 units while the second with the number of individuals are applied to the end of each network producing a logit vector for each network. The scale

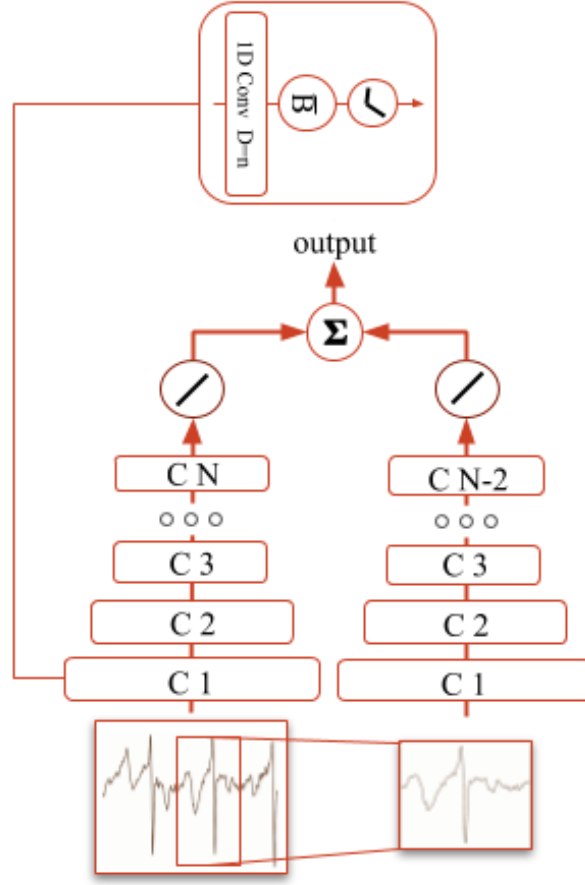


Figure 8.18: Proposed TCNN architecture.

differences between the input used by each network and the progressive dimensionality reduction of the TCNN generate variability in the number of layers. For the Fantasia and MIT-BIH databases, the number of convolutional layers is 6 non-fiducial network and 4 for the individual cycle network. Due to the increase of sampling frequency of the CYBHi database, the number of layers is adjusted to 8 and 6, respectively.

Each of these networks is trained independently using a cross-entropy loss function and Adam optimizer. After trained, the logit vectors from both networks are fused with the sum rule. The hypothesis, in this case, is that the higher the value in each position of this array, the higher the probability of being the individual. The score, in this case, is given by subtracting one to the normalized output vector (o_n), given by:

$$S(p, i, w) = 1 - \frac{o_p}{\max(o(i, w))} \quad (8.3)$$

8.3.7 Relative Score Threshold Classification

The relative score threshold classifier is a method that classifies choosing the lowest normalized similarity score ($\bar{S}(p, i, b)$) for each batch of windows (b), consequently, S is a 3

dimension tensor with dimensions $(M \times I \times B)$, where B is the size of the batch. This measurement is obtained by normalizing for each predictor the minimum of a set of windows. Therefore the minimum is the result of the following function:

$$S_{p,i,b} = \min \left[\left\{ S_{p,i,B-b}, S_{p,i,B-b+1}, \dots, \bar{S}_{p,i,B-b+B} \right\} \right] \quad (8.4)$$

and the normalization for each b in relation to the p is made with the following rule:

$$\bar{S}_{p,i,b} = \frac{S_{p,i,b} - \min(S_{\mu,i,b})}{\max[S_{\mu,i,b} - \min(S_{\mu,i,b})]} \quad (8.5)$$

, where μ is the group of all predictors in the batch of windows. Considering that $\bar{S}(p, i, b) \in [0, 1]$ the value 0 encodes to the most probable class and when closer to 1, the lower probability. The class of the individual is given by $C_{i,b}^I$:

$$C_{i,b}^I = \arg \min_p [\bar{S}_{\mu,i,b}] \quad (8.6)$$

This multimodal classifier is used for identification and its evaluation is given by the accuracy, specificity, and sensibility.

When comparing each score of all predictors of a certain b window from a specific i with a changing threshold (φ), one can design a binary classification system ($C_{p,i,b}^A$) for each individual. This system will output if that specific b of windows belongs to claimed individual:

$$C_{p,i,b}^A = \bar{S}(p, i, b) > \delta, \quad C^V \in 0, 1 \text{ and } \bar{S} \in [0, 1] \quad (8.7)$$

The Receiver Operating Characteristic (ROC) can be calculated with an array of those thresholds (Fig. 8.19) and the Equal Error rate (EER) is calculated. The authentication mode is evaluated with this measurement, as these systems prioritize the minimization of the possibility of imposters that can access the system and the number of the rightful individuals that cannot log in.

8.3.8 Results

This section will present the results for each database, ordered in increasing average intra-variability across subjects. Apart from the validation for identification and authentication modalities, the Fantasia dataset will be the benchmark to test the used methods and robustness of the proposed algorithms.

8.3.8.1 Fantasia

Of the first hour of the Fantasia dataset 1/3 of data was reserved for training and 2/3 for testing. The first part was used for testing and all data was segmented in windows of 512 samples (approximately 2s) with an overlap of 67%. The selected windows were on

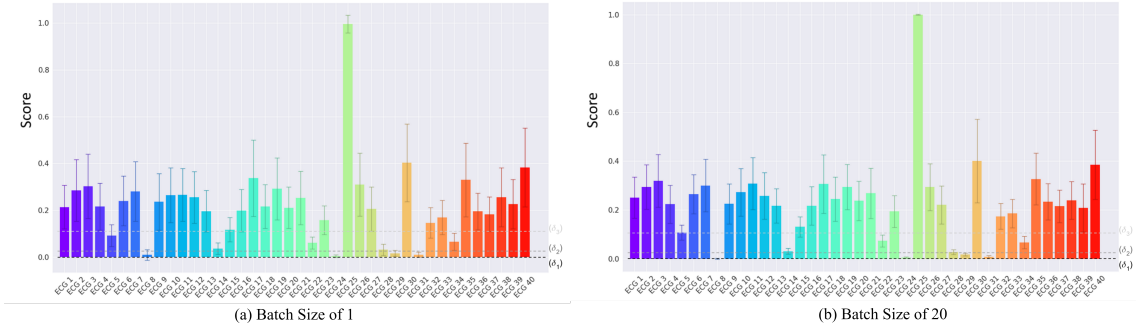


Figure 8.19: Distribution of the score of the RNN network for the predictor *ECG 8* for each of the ECG signals of the Fantasia dataset. While (a) refers to a single window of 512 samples, (b) is using the evaluation for a batch size of 20 windows. δ_1 , δ_2 and δ_3 are three examples of increasing threshold values.

average 1464 windows of 3787 per subject (aproximately 39%). The rejection algorithm had a mean and standard deviation tolerance of 0.2 and 0.05, respectively.

The score distribution per subject when the test set is fed into the RNN model trained with *ECG8* is depicted in Fig. 8.19. The score in the right is given for all batches of windows for $B = 1$ and the left for $B = 20$.

When analyzing this example, it can be observed that with the increase of the number of windows included in each batch, both mean and variability are reduced. The three thresholds presented in Fig. 8.19 will have a different binary classification for the predicted *ECG₈* when submitted to different values of B . In the case of (a), δ_1 threshold will give a positive value for most of the *ECG23*, which is false, but both δ_2 and δ_3 will include the *ECG₈* batches as well, but also the number of imposters increase with the threshold value.

When the B increases to 20, the variance and mean value of *ECG 8* will decrease significantly and δ_1 will now classify correctly most, or even all, the *ECG 8* batches, while reducing the number of *ECG23* classified wrongly, ensuring lower values for the False Negative Rate (FNR) and False Positive Rate (FPR). Both δ_2 and δ_3 produce more imposters, while not increasing the correct classification. The ROC curve is generated for each predictor and each batch size, producing different values for EER that can be interpreted for authentication systems.

The identification rate, i.e. accuracy for the identification modality, for both algorithms are depicted in Fig. 8.20. Close inspection reveals that both algorithms increase accuracy with time per batch, but the TCNN algorithm starts with higher accuracy, being outperformed by RNN after approximately 1 minute of signal per batch. While the RNN algorithm increases until reaching 100 % at approximately 112 s, the TCNN algorithm reaches 99.1% after approximately 90 s, but its curve displays a stabler behavior. The best results are presented in the confusion matrices of Fig. 8.21 and 8.22 for RNN and TCNN approaches respectively.

The results for the authentication mode is depicted in Figs. 8.23 and 8.24, which

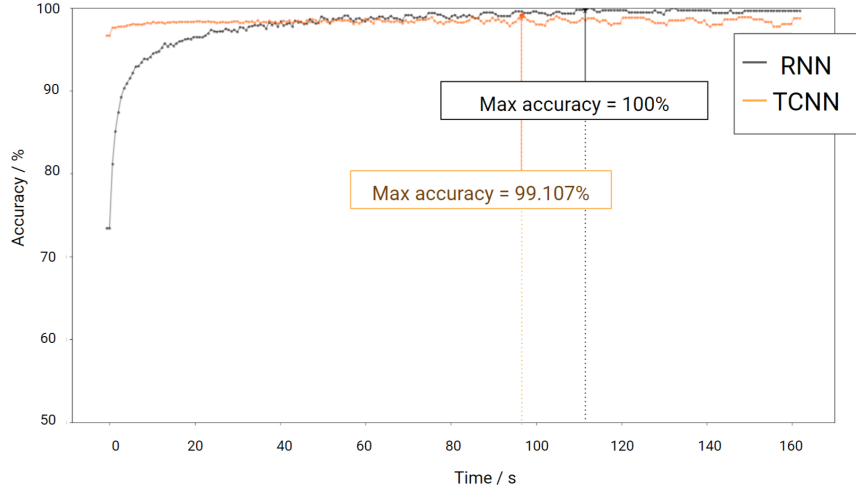


Figure 8.20: Results for identification on the Fantasia dataset with the increase of batch sizes.

shows the evolution of the EER per time in each batch. Both RNN and TCNN achieve values very close to 0% at 80 s, but the TCNN reaches those values with a lower standard deviation faster than the RNN counterpart.

Due to the low EER scores provided by these algorithms, their robustness to Gaussian noise, with decreasing Signal-to-Noise Ratio (SNR), was tested. As the default SNR calculated for Fantasia was 15 ± 3 dB, Fig. 8.25 shows the impact on the EER with the progressive decrease of SNR from 12dB to 7dB. In this experiment, both algorithms were trained and tested with the corrupted ECG, for all the 3587 windows per subject. As expected, both architectures suffer from the increase of noise because the signal turns more unpredictable and more random with the Gaussian noise. One unexpected fact was for the SNR of 11dB for the RNN strategy, as it had an increase of EER in comparison to the 10dB. It can be seen that the robustness for SNR of 10 and 12, reach EER values lower than 1%. The TCNN strategy proves to be more robust to Gaussian noise as it only achieves an EER above 1% when the SNR is lower than 7. It can also be seen that the behavior of the TCNN algorithm also differs, as the increase of B does not reflect a significant difference in results.

8.3.8.2 MIT-BIH

All the MIT-BIH signals were resampled to a frequency of 250 Hz to ensure that the same sampling frequency is maintained in all datasets for a rigorous comparison. Like the Fantasia dataset, each record was divided by 1/3 for training and 2/3 for testing. While using the mean and standard deviation window removal algorithm 42% of the time windows were rejected with the parameters for tolerance for average and standard deviation of 0.9 and 0.5, respectively.

The identification test performed in the MIT-BIH database, observed in Fig. 8.27,

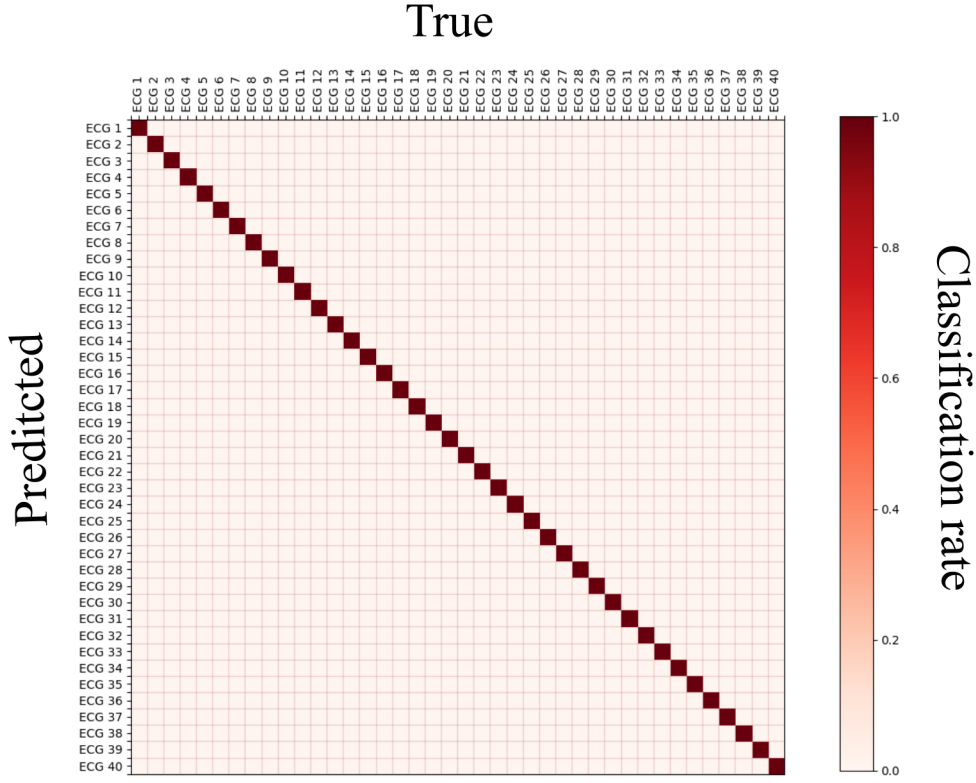


Figure 8.21: Confusion matrix for the classification of the identification of each individual of the filtered for Fantasia and RNN for the best accuracy results. Accuracy of 100%, specificity of 100% and sensitivity of 100%

shows that the accuracy curves have similar behavior to the ones in the Fantasia dataset. For the sake of comparison with literature, an analysis of the normal sinus rhythm (NSR) dataset was also made in detail (Fig. 8.27). Even though the RNN approach does not go above 93% for all the MIT-BIH and the confusion matrix (Fig. 8.28) shows the expected diagonal. Unfortunately, some models generalize the ECG signals confusing the right class, probably due to the loss of information during resampling, due to the sensibility to noise or to due to the symptomatic behavior of the arrhythmia events. The last option is supported by the fact that the NSR subset reached 100% accuracy (Fig.8.27).

As for the TCNN even though the 96.4% is reached after more than 2 min for the full MIT-BIH database, one can observe that values close to 96% are reached after 10 s, reaffirming its robustness to noise and stability. The confusion matrix in Fig. 8.29 shows that only two predictors display a faulty behavior, probably due to the presence of arrhythmia events. When comparing the individuals from the NSR the accuracy reached 100% after approximately 10 s (Fig. 8.27).

The authentication evaluation of all the MIT-BIH individuals is displayed in Figs. 8.30 and 8.31. This figure shows that the TCNN approach reaches EER values close to 0% with 72 s in each batch and 1.7% for RNN. Even though the best result for TCNN is after

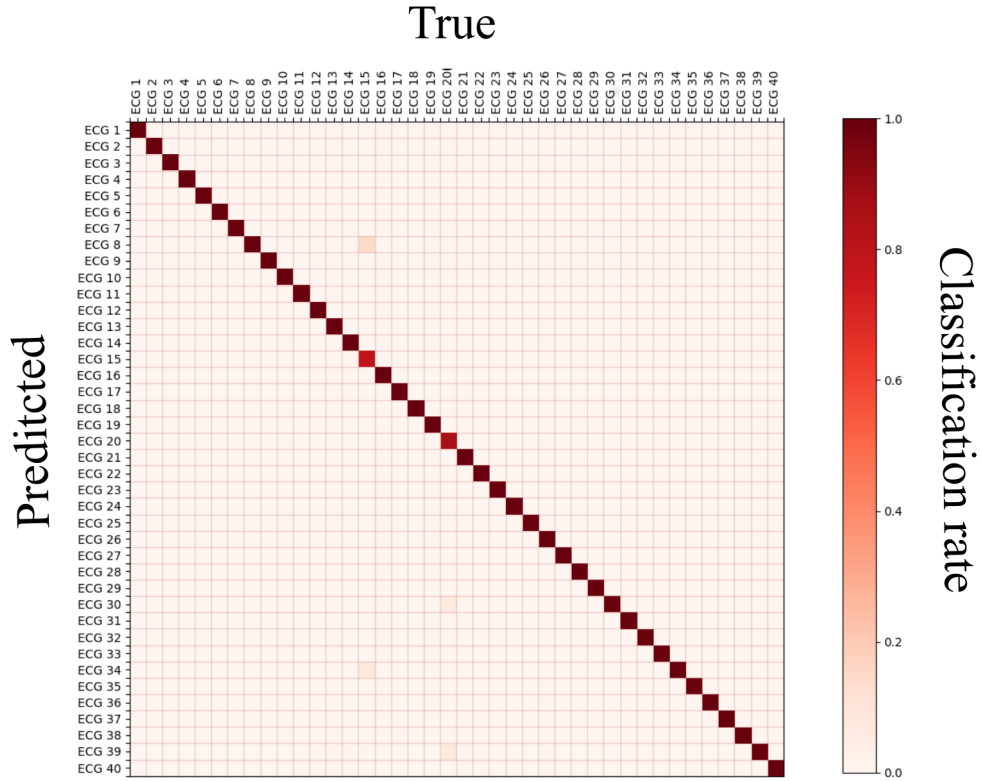


Figure 8.22: Confusion matrix for the classification of the identification of each individual for Fantasia and TCNN for the best accuracy results. Accuracy of 99.1%, specificity of 100% and sensitivity of 99.1%

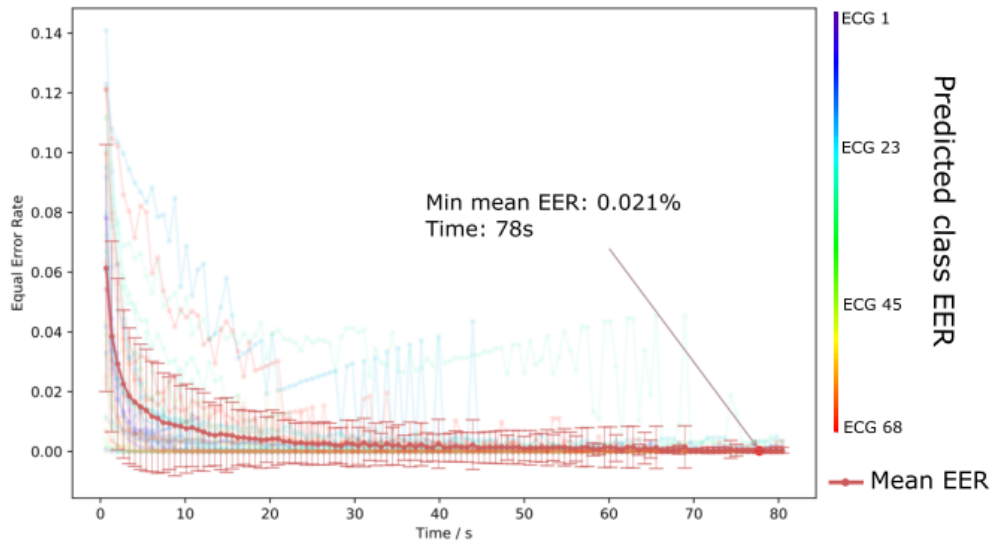


Figure 8.23: Evolution of the EER values over batch widows duration for the Fantasia dataset of the RNN model. The error bars represent the standard deviation for all the EER distribution.

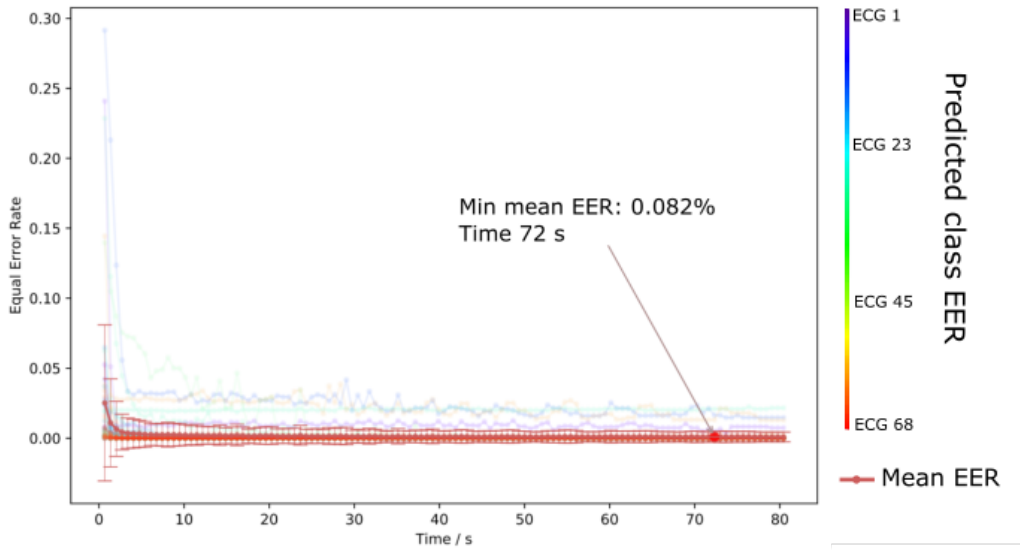


Figure 8.24: Evolution of the EER values over batch widows duration for the Fantasia dataset for the TCNN model.

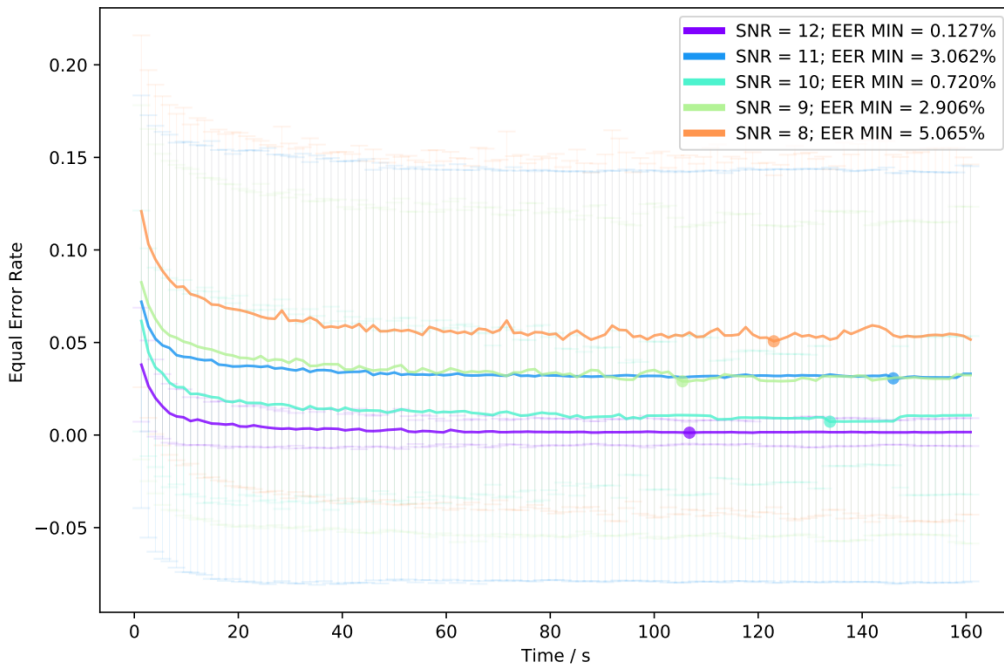


Figure 8.25: Evolution of the mean EER values over batch widows duration for the Fantasia dataset with added Gaussian noise for the RNN method.

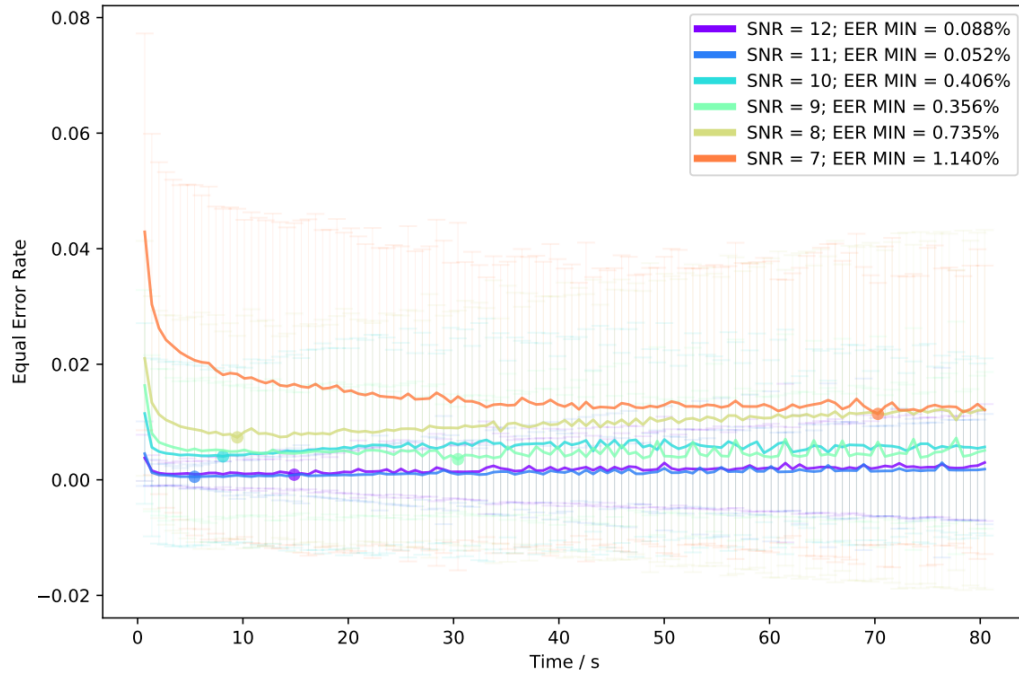


Figure 8.26: Evolution of the mean EER values over batch widows duration for the Fantasia dataset with added Gaussian noise for the TCNN method.

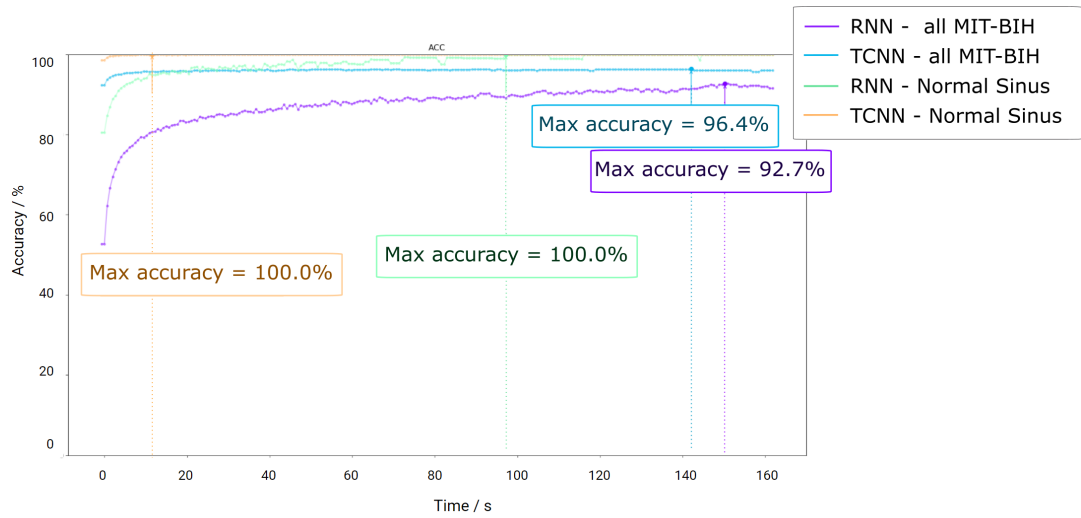


Figure 8.27: MIT-BIH identification accuracy

1 min, the values are low from the start. For the NSR database (Fig. 8.31) the TCNN reached 0% of EER, while the RNN reached the value of 0.6% (Fig. 8.30).

8.3.8.3 CYBHi

As mentioned before the CYBHi database comprises two moments with a time distance of two months. Therefore, both algorithms are evaluated with all the combinations of

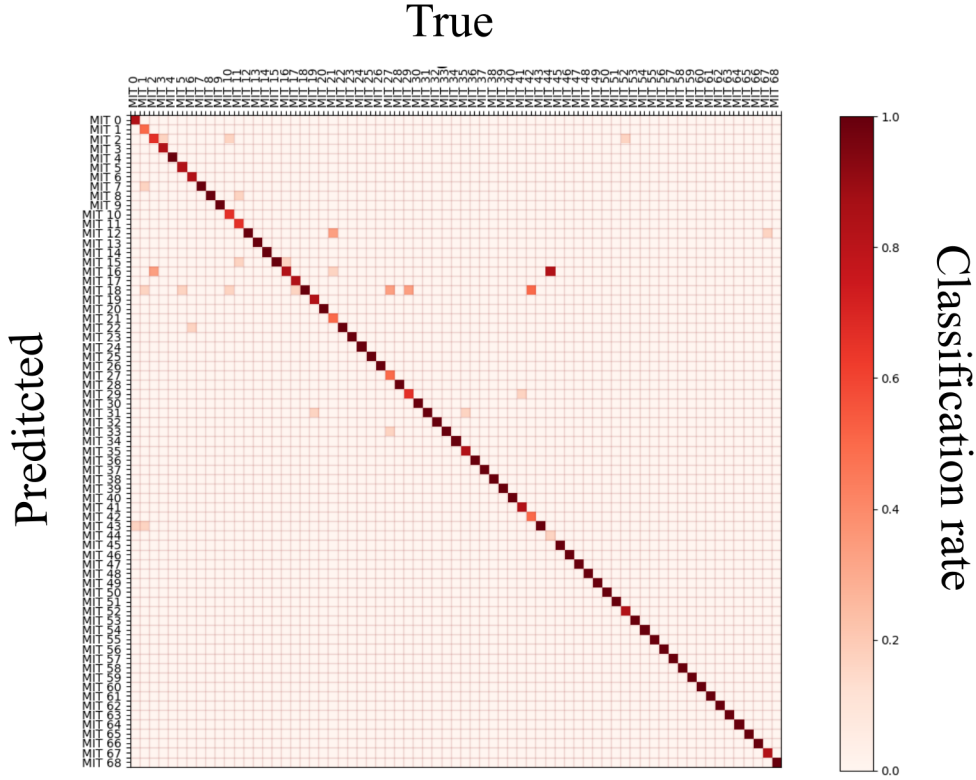


Figure 8.28: Confusion matrix for the best results for MIT-BIH with the RNN approach. Accuracy: 92.7%; Specificity: 99.9% ; Sensitivity: 92.7%.

both moments (M1 and M2) in terms of cross-validation, simulating the enrolment and verification environments of a biometric system. This means that each model was trained using the first element and tested with the second of the following sets: "M1 vs M1", "M2 vs M2", "M1 vs M2" and "M2 vs M1". When the training and testing moment was the same, the data was separated by 50% and the segmentation was made with an 89% of overlap to increase the number of available time-windows.

The clustering method for the noise detection rejected about 9% of the total signal length on the first session and 10% on the second.

In Fig. 8.32 is shown that the decrease of accuracy reflects the signal corruption that this dataset faces. Since this dataset displays a high variability within subjects due to the low SNR, the robustness of the method is imperative for good results. The TCNN algorithm archives results above 90% for "M1 vs M1" and 100 % for "M2 vs M2", even with an amount of information that was significantly lower than the previous datasets. The decrease on accuracy of crossing both moments as significant as expected, revealing accuracies above 75 % for both "M1 vs M2" and "M2 vs M1". The low accuracy presented by the RNN approach reflects the previously observed sensitivity to noise.

Fig. 8.33 shows that the EER archive 4.3% and 6.1% for "M1 vs M1" and "M2 vs M2", respectively, using the RNN models, while TCNN displays results close to 0%, probability

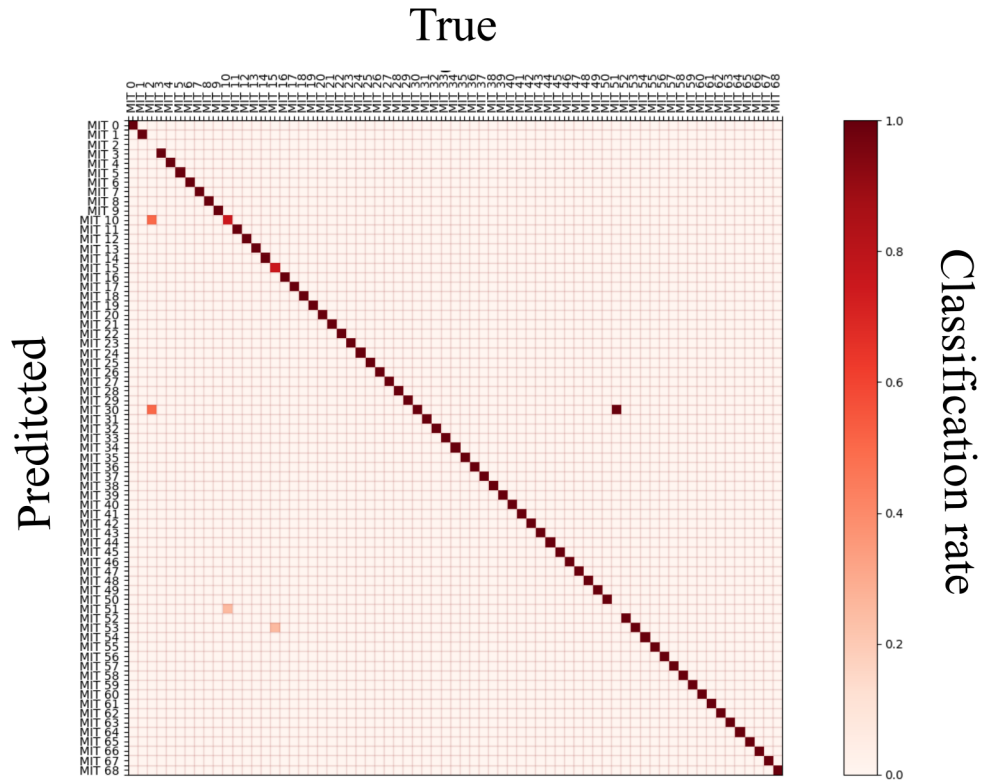


Figure 8.29: Confusion matrix for the best results for MIT-BIH with the TCNN approach. Accuracy: 96.4%; Specificity: 99.9%; Sensitivity: 96.4%.

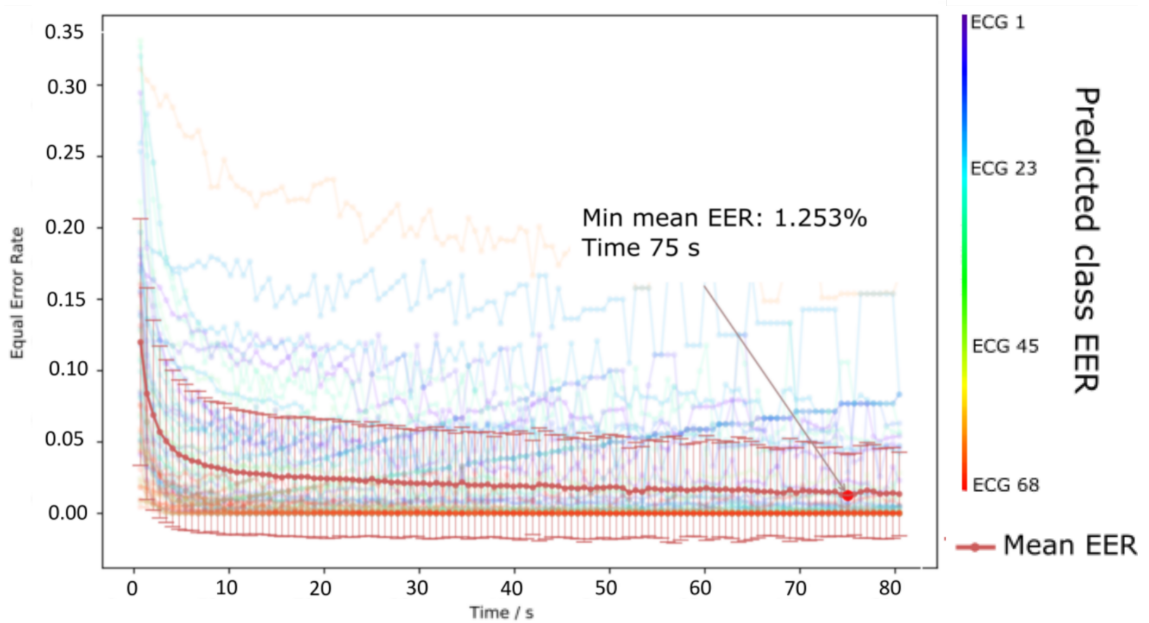


Figure 8.30: Evolution of the EER values through time for the MIT-BIH dataset for the RNN method.

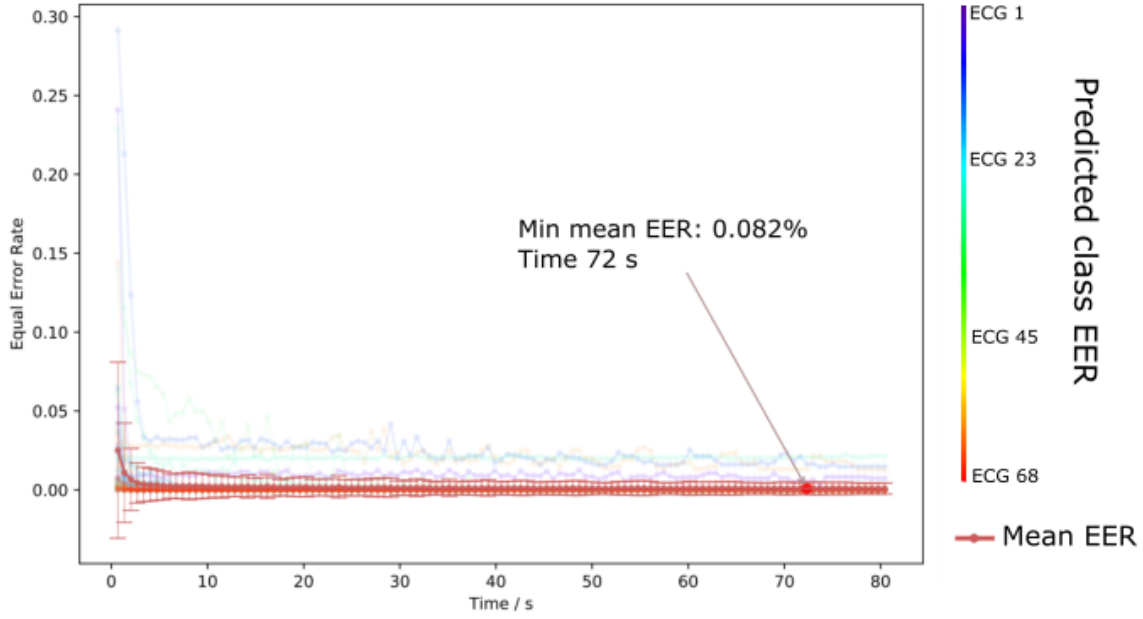


Figure 8.31: Evolution of the EER values through time for the MIT-BIH dataset for the TCNN method.

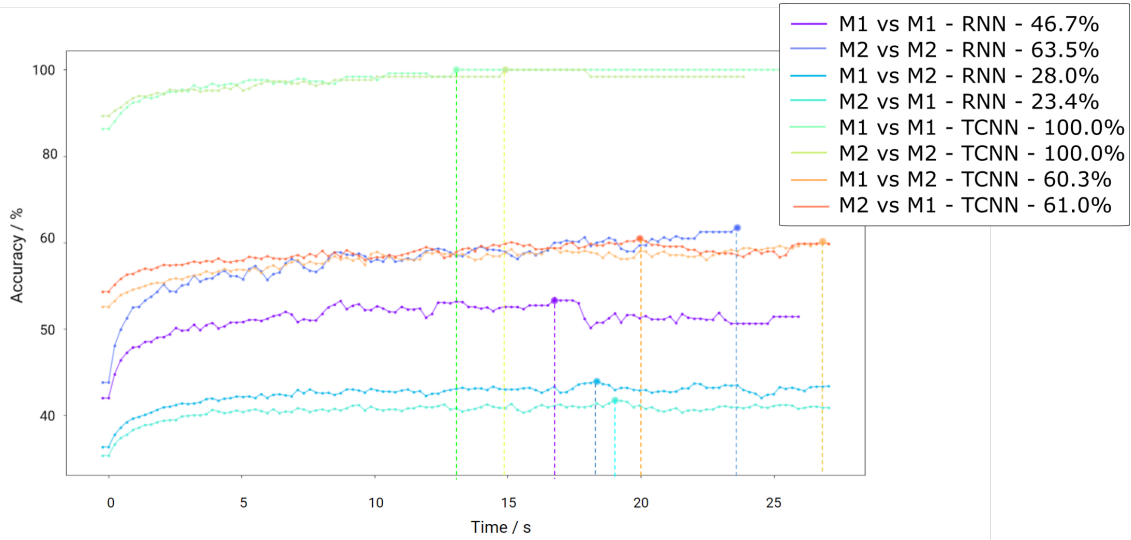


Figure 8.32: CYBHi identification accuracy. M1 represents the first moment and M2 represents the second moment.

due the temporal proximity of the training and testing sets. As for the evaluation across different moments, the EER for the RNN approach is 18.3% and 18.8% for "M1 vs M2" and "M2 vs M1", respectively and the 2.2% and 4.1% for the TCNN in the same order. These results, even though they are based in a higher quantity of information, they outperform current state-of-the-art for off-person measurements.

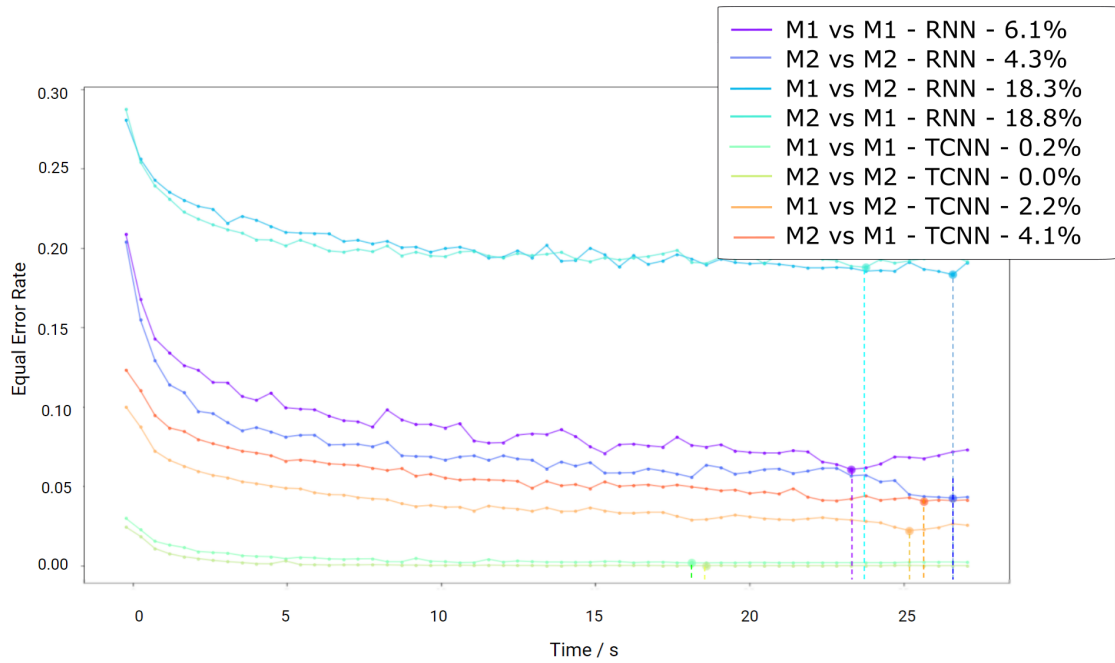


Figure 8.33: CYBHi authentication EER values. M1 represents the first moment and M2 represents the second moment.

8.3.9 Comparative Study

Table 8.4 summarizes the results per database for the consulted bibliography. This table shows that the TCNN approach outperforms most of the other studies in both identification and authentication paradigms. The robustness of this architecture is mainly due to the combination of bringing together both fiducial and non-fiducial machine-learned characteristics of the signal through the fusion layer.

Some challenges for this algorithm were detected, namely due to the computational limitations. Therefore in order to provide a biometric system in a real scenario, the following items should be tackled:

- The increase of samples per window also increase the needed computational power for training. One possible solution would be to have a state-of-the-art downsampling mechanism with the minimum loss of information possible;
- The increase of number of people would make the network heavier to process. Even though the training could be performed in batches of people, and even do transfer learning, from a generic ECG database, to decrease the amount of data needed from each person;
- when a new person is added to the model, all training must be repeated. One solution would be to use the RNN method that does not have this problem. Other solutions would be to create a network with memory and number of users higher than the ones that exist, so that when new people are added, the network only needs to fill the

Table 8.4: Comparison with Previous Work

Study	Database	Acc	EER
Tantawi, Revett, and Salem (2013) [160] Tantawi et al. (2013) [161] Gargiulo et al. (2015) [162] RNN TCNN	Fantasia	90.0% 95.9% 99.0% 100% 99.1%	— — — 0.02% 0.02%
Wang et al. (2008) [208] Fatemian and Hatzinakos (2009) [209] Wang et al. (2008) [208] Rabhi and Lachiri (2013) [164] Sidek, Mai, and Khalil (2014) [165] RNN TCNN	MIT-BIH Sinus	98.1% 99.6% 94.5% 99.0% 99.1% 100% 100%	— — — — — 0.6% 0.0%
RNN TCNN	MIT-BIH Arr, Sin, Long	92.7% 96.3%	1.2% 0.1%
Da Silva et al. (2014) [167] Lourenço, Silva, and Fred (2012) [168] da Silva Luz et al. (2018) [102] RNN TCNN	CYBHi M1 vs M1	94.4% 95.2% — 46.7% 100%	— — 1.3% 6.1% 0.2%
Lourenço, Silva, and Fred (2012) [168] da Silva Luz et al. (2018) [102] RNN TCNN	CYBHi M1 vs M2	90.2% — 28.0% 60.3%	— 12.8% 18.3% 2.2%
da Silva Luz et al. (2018) [102] RNN TCNN	CYBHi M2 vs M1	— 23.5% 61.0%	14.0% 18.8% 4.1%
RNN TCNN	CYBHi M2 vs M2	63.5% 100%	4.3% 0.0%

prepared gaps. Another solution would be to train more than one network, each one with each set of people and use the same method of classification;

- this approach is viable for a suitable number of individuals, but to use in a scale of millions, the training and testing process will be costly or even impossible with state of the art computational power and memory. This challenge could be overcome with the creation of more than one network, each for a set of people and subject them with the same classification method;
- Another disadvantage is the need to extract the peak information, which sometimes is not feasible, especially when the slow waves (such as T or U) reach higher values than the R wave. This challenge could be made by using other state-of-the-art mechanisms to find the peaks, namely using the Syntactic Search in Time-Series (SSTS) method described previously.

Even though the RNN achieves state of the art results for the Fantasia and NSR Rythm of the MIT-BIH, the sensibility to changes in the signal is higher than the aforementioned architecture. The most prominent possibilities for this sensibility is:

- The noise may have a high impact in the quantization process, especially due to random artifacts that shift the maximum and minimum values of the signal;
- The loss of information due to the preprocessing stage caused by the decimate method, which could be important in establishing the difference between individuals.
- The need for higher quantities of data for training, which could contain more angles of the ECG mechanism.

Another limitation for this system is the required computational power, due to its recursive nature, it is not possible to parallelize between sequential samples, leading to the training of 512 windows with 512 samples an average of 35.5 hours with an Nvidia GTX 1080 Ti. In terms of testing, the score calculation for each window is just on the scale of approximately 10 milliseconds.

The main advantages of the RNN architecture are:

- It is unnecessary to train all the models again when adding a new person to the biometric system as the individual traits of one individual would be preserved in a single model that is independent of the others;
- The non-fiducial approach removes the need to extract the R peak from the signal;
- This approach achieves excellent results on signals with good SNR, therefore this system is viable for good quality acquisitions.
- The possibility of generating the signal using the same method as in section 8.2 to observe the differences with the individual, to track models that did not learn correctly.

8.4 ECG Abnormality Detection

As previously mentioned, the morphological and spectral components of the ECG cycle provide hints of the emotional, behavioral and health state of the individual. In the context of medicine and healthcare in general, the analyses of the changes in cardiovascular activity can be accessed easily through wearable devices. Automatic detection by using signal processing and pattern recognition methods may provide an asset in providing relevant information without having to rely on hospital resources [210, 211].

With a better understanding of cardiac pathologies and the development of diagnostic methods and therapies allied to the evolution of technology in the medical field, greater healthcare expectations emerged in terms of efficiency.

The hypothesis we conjure is that as a DNN learns the default morphology of a normal sinus rhythm signal it will also be trained to be capable of analyzing and detecting its divergence caused by a modified morphology. On that account, we propose a framework that detects unexpected changes in two different scenarios: contamination due to noise and artifacts, and; the presence of symptomatic occurrences. Each scenario will have its architecture, but they will provide a first step in assisting medical professionals in detecting, not only, the quality of the signal, but also, the presence of arrhythmia events.

8.4.1 Methods

To conduct this experiment, four sequential preprocessing methods were used:

1. Decimation - reduce the computational burden while increasing the time frame displayed per window;
2. Normalization - using the "min-max" equation (5.5);
3. Average removal - for faster training convergence;
4. Segmentation - the signal will be segmented in to 64 samples per window, equivalent to 1 second.

To perform ECG classification according to the two tasks at hand, two different architectures were constructed and optimized to each considered problem. Since autoencoders perform well in learning general features from the input, they were selected as the base for the implementation of both algorithms.

In the first stage, an autoencoder will learn the characteristics of a normal ECG signal in the purest form possible, i.e. with a good SNR and no symptomatic events. Consequently, the latent vector will contain the coded information of the input, and also called "feature vector". When this structure is fed with corrupted ECG the algorithm will produce a different latent space, changes that can be analyzed and detected using a classifier.

The training of each algorithm was performed using batches of 256 shuffled windows, each without overlap for the noise detection experiment, and 50 % of overlap for the arrhythmia experiment. The testing was made with the rest of each signal. Since the normal ECG cycle occurs during one second it is expected that each window contains at least one QRS complex.

8.4.1.1 Autoencoder

The diagram for the encoder is depicted in Fig. 8.34, which is comprised of three convolutional blocks. Each block contains one convolutional layer with ReLU activation and max pooling. In the end, the information is transformed and encoded into a latent vector with 256 elements.

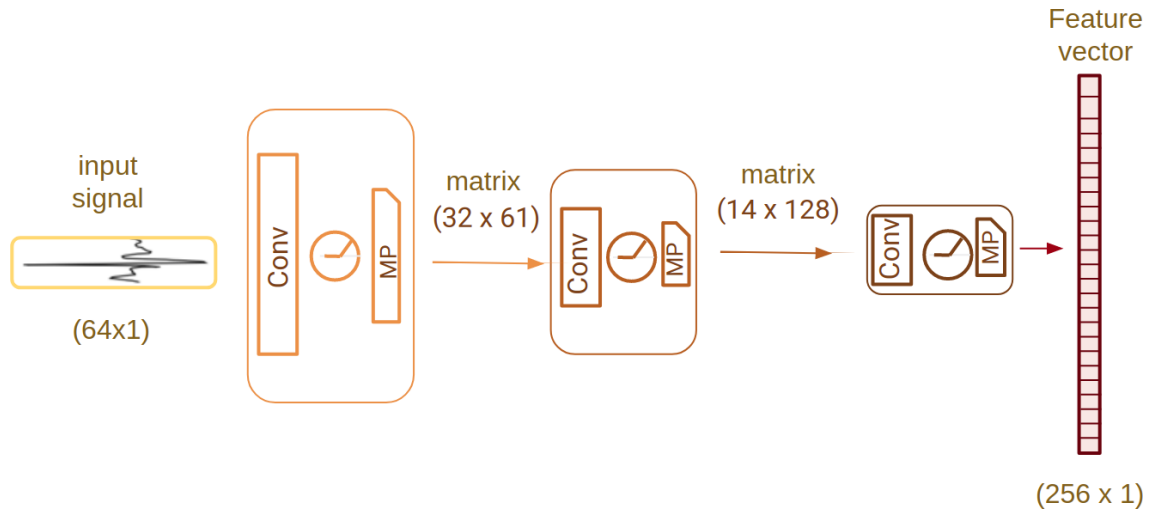


Figure 8.34: The encoder is composed by three blocks composed by a convolutional ("Conv") with a ReLU activation and a max pooling ("MP") layers.

The training, monitoring, and optimization of the encoder is made by training an autoencoder, using the MSE as the loss function and RMSProp as the optimizer (initial learning rate of 0.001). After training, the decoder module will be discarded, as the purpose is to use the feature vector produced by the encoder module.

8.4.1.2 Noise Detection Network

The noise detection network is the architecture that will be used to detect the differences between the normal signal (NS) and different types of noise. At first, a binary noise detection model was developed that is capable of classifying between noise affected signal (NAS) or NS. Then, a second model was developed to detect three types of corruption commonly found in ECG segments:

- baseline wander (BW) - artifacts are caused by movement such as the contraction and relaxation of the thoracic cage muscles, which is evident for periodic events;
- electrode motion (EM) - originated from the misplacement and/or loss of contact between electrodes and the skin;
- muscle artifacts (MA) - caused by the interference of EMG signals during the activation of the muscles caused by the movement of the human body.

Both algorithms follow the same structure, as depicted in Fig. 8.35, with only the difference between the number of layers in the fully connected network. As the network progresses toward the end, each layer decreases by half the number of a neuron, starting in 256, with ReLU activations in the first layers and softmax in the last one. Since the last layer contains the number of classes for each system, the first settled with 9 while the second with 10 layers.

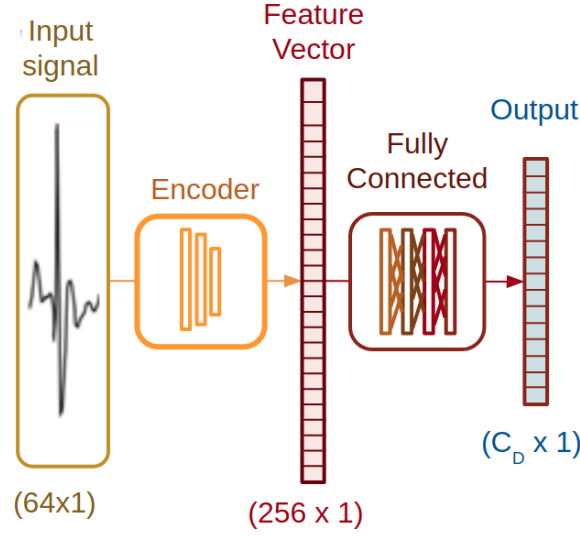


Figure 8.35: Model architecture for the noise detection, with C_D classes.

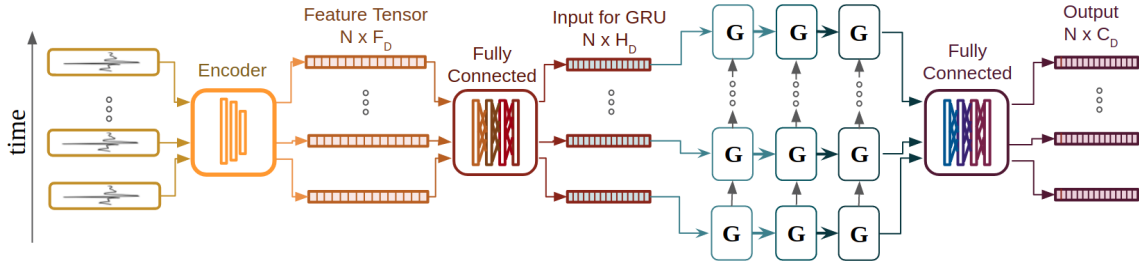


Figure 8.36: Model architecture for the detection of different types of arrhythmia ($C_D = 7$).

In both setups, a dropout of 50% is implemented in the fifth layer, to obtain a faster convergence and produce better results. The training of this module was also made with the RMSProp optimization method with cross entropy loss function.

8.4.1.3 Arrhythmia Detection Network

To include a sequential approach for detection and prevention of arrhythmia events, portrayed in the diagram in Fig. 8.36, it resorts to the use of an encoder, dense layers and a RNN comprising three sequential GRU layers.

After ECG features from each widow are extracted using the previous CNN encoder, N number of feature vectors are stacked, resulting in matrices of size $(N \times F_D)$, where F_D is the dimension of the feature vector. Since the RNN module has its own set of hidden units (H_D), the fully connected layer, with linear activation, makes the connection between both modules. The parameters used for this experiment were $F_D = 256$ and $H_D = 128$ and $N = 32$.

As mentioned before the RNN module contains three GRU layers and the output of the last GRU layer is then carried into the last fully connected layer, which ultimately classifies for six different classes of arrhythmia events and one for normal ECG (NS) ($C_D = 7$).

The final fully connected block starts of with a batch normalization layer and afterward the same sequential layered system, as used in the noise detection algorithm, is applied. In this instance, the first layer has the size of H_D , in each layer the size decreases until reaching the size of C_D , consequently the final layer has 7 units instead of the expected 8.

8.4.1.4 Training

The training of this algorithm was made with RMSProp with the initial learning rate of 0.001, which was automatically decreased abruptly in the course of the training phase, according to three parameters, to ensure the maximum convergence:

1. If the loss value doesn't change for a certain amount of iterations;
2. If the mean of the lost value in the last 10 iterations increases fifteen times (arbitrary value);
3. If the relative loss is bellowed the value of 10^{-4} .

The training stopped once one of the rules was broken five times in a row, or if the learning rate was bellow 10^{-6} .

The input data is separated into batches of 256 signal windows that were shuffled so that the models could reach better generalizations while avoiding overfitting. This method ensures that training was within the computational limitations.

8.4.2 Datasets

To train the autoencoder, it was important to choose a dataset that contains a good SNR and without other abnormal morphologies, therefore the Fantasia dataset was chosen for this task.

For the noise detection model, apart from the clean signals compiled in the Fantasia database, MIT-BIH Noise Stress Database. This last database contains the labeled segments for the classes of EM, BW and MA.

For the last task, the MIT-BIH Arrhythmia database was used for the considered classes: Atrial Fibrillation (AFIB), Atrial Flutter (AFL), Ventricular Bigeminy (B), P, Pre-excitation (PREX), Sinus Bradycardia (SBR), and NSR.

8.4.3 Results

All models were developed, trained and tested using Keras API on top of Tensorflow and the used hardware included an NVIDIA GeForce GTX 960 GPU. The results will be organized by each sequential experiment.

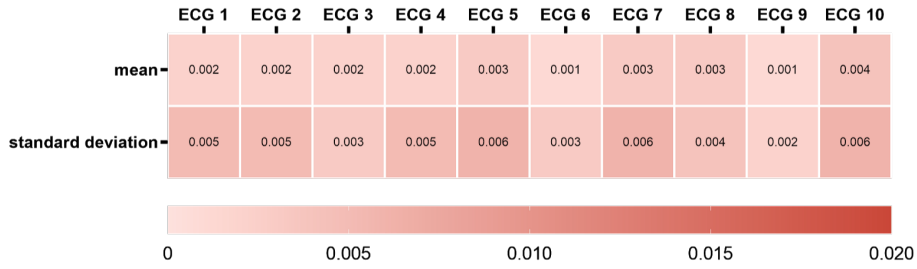


Figure 8.37: MSE mean and standard deviation values for the first 10 subjects within the age group of 21 to 34 years old.

8.4.3.1 Autoencoder

After approximately 2100 epochs of the training phase, the minimum reached MSE was 0.0001. For the testing phase, a portion of every signal available in the Fantasia Database was used, while being careful not to overlap with the training data. The MSE was calculated for each signal reaching a mean value of 0.0026 and a standard deviation of 0.0012. The results for the first 10 individuals of the dataset, which have ages between 21 and 34 years old, are depicted in Fig. 8.37.

The low values exhibited by the results suggest that the model was able to reconstruct the main characteristics of the signal. The signal depicted in Fig. 8.38 presents one example of the reconstruction capabilities of the autoencoder. As seen in this graphic, the model was able to mimic the same apparent frequency, baseline values, and fluctuations as the original signal of the ninth individual of the dataset.

After analyzing the results, it was concluded that the encoder could be applied for the following classification models.

8.4.3.2 Noise Detection Neural Network

As stated before, the binary noise detection model classifies ECG segments according to two classes: NS or NAS. The training proceeded until 600 epochs, where an abrupt increase of error was observed, reaching a minimum cross entropy of 0.51 with an accuracy of 98.6%.

The classification results, shown in Table 8.5, provides evidence that the model was able to successfully detect the presence of noise in the ECG. Further support for this claim is presented in the confusion matrix exhibited in Fig. 8.39.

Table 8.5: Binary noise detection model: classification performance (%)

Accuracy	Sensitivity	Specificity
98.18	98.21	98.15

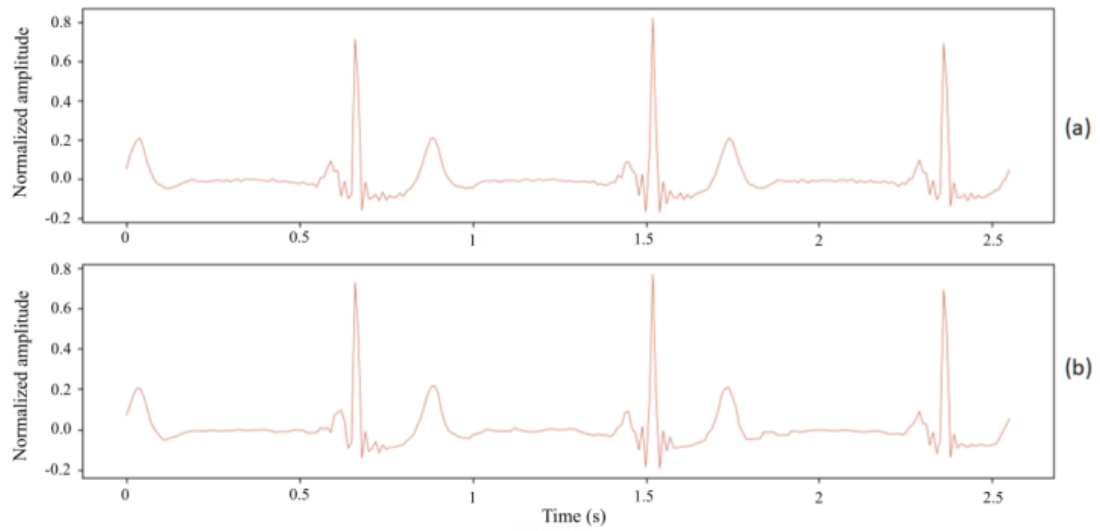


Figure 8.38: Portion of the signal ECG 9, from the Fantasia database (a) and reconstruction of the same signal (b).

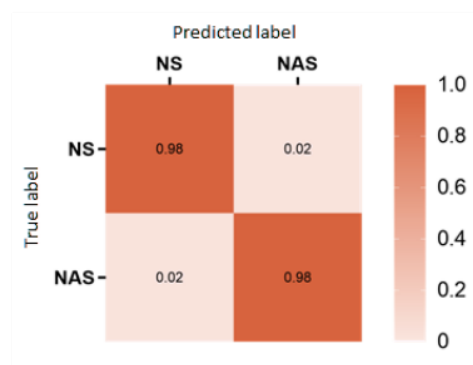


Figure 8.39: Normalized confusion matrix of the dataset used for the training phase, where Normal Sinus (NS) is the positive label and Noise Affected Sinus (NAS) is the negative label.

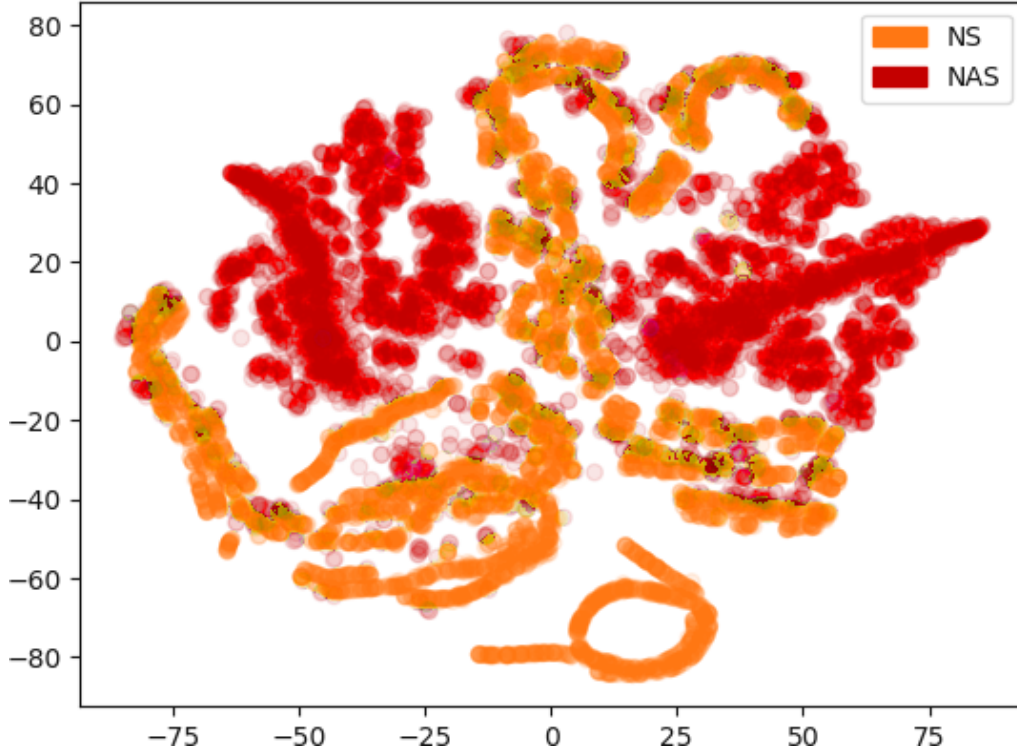


Figure 8.40: t-SNE representation of Normal Sinus (NS) (orange) and Noise Affected Sinus (NAS) (red) feature vectors produced by each window encoding.

The encoded data for both classes were submitted to t-Distributed Stochastic Neighbour Embedding (t-SNE) to visualize the proximity between the feature vectors. Therefore, each point in the resulting graph represents the tensor created by the encoder originated by a single-window input. These results, seen on Fig. 8.40, show a good separation between the NS and NAS data points, each represented by a color. This confirms that the encoder extracts different values for each feature represented by the two classes.

As for the multi-level noise detection algorithm the ECG segments can be classified according to the following classes: Electrode Motion (EM), Baseline Wander (BW) and Muscle Artifacts (MA). During the training phase, a MSE of 0.85 was reached after 600 epochs with an accuracy of 86,2%. However, the testing accuracy only achieved by 70.7%.

The evaluation of this model was taken into account the attribution of positive to a given class while giving negatively to the rest. The results for each class are conveyed in Table 8.6.

Table 8.6: Multi-class noise detection model: classification performance for each class (%)

Class	Sensitivity	Specificity
NS	89.77	96.63
BW	72.70	84.26
EM	65.19	91.35
MA	59.11	89.47



Figure 8.41: Normalized confusion matrix of the dataset used for the testing phase, with classes Normal Sinus (NS), Electrode Motion (EM), Baseline Wander (BW) and Muscle Artifacts (MA).

By inspecting the normalized confusion matrix in Fig. 8.41, it can be stated that even though the algorithm can correctly identify normal signals most of the time, it is harder to correctly separate the identity of each type of noise.

Comparing these results with the previous experiments, we conclude that even though it is possible to detect noise corrupted ECG with high accuracy, but differentiating between the several types of noise reveals to be a harder task. This may be because most of the time signals are affected by different types of noise sources at once and the fact the encoder was not trained to deal with these differences.

8.4.3.3 Arrhythmia detection model

The training phase stopped when the minimum cross entropy reached 0.65 and on accuracy of 90.7% after 1750 epochs. The accuracy dropped to 56.9% for the test data.

The confusion matrix, displayed in Fig. 8.42, shows that the model was able to correlate between certain characteristics with the correspondent arrhythmia, but underperforming between distinguishing some of them. This DNN architecture had a high classification accuracy rate for the P in contrast with the detection of AFIB, often confusing its morphology with AFL since both of them are quite similar.

These results are promising and suggest that the development of this algorithm by fine-tuning its parameters and have some considerations in the training phase of the autoencoder. Unfortunately, at this current stage, it cannot be used in a real-life context, since the model often considers pathological ECG segments as NSR, giving too many false positives and negatives. By analysing the classification performance of each class presented in Table 8.7, it is possible to conclude that the algorithm often struggles to correctly identify ECG segments with B, AFL and AFIB, reaching an average sensitivity and specificity for this model of 61.1% and 93.1%, respectively. But we are confident that these results can be improved with more powerful computation hardware and more neural architectural

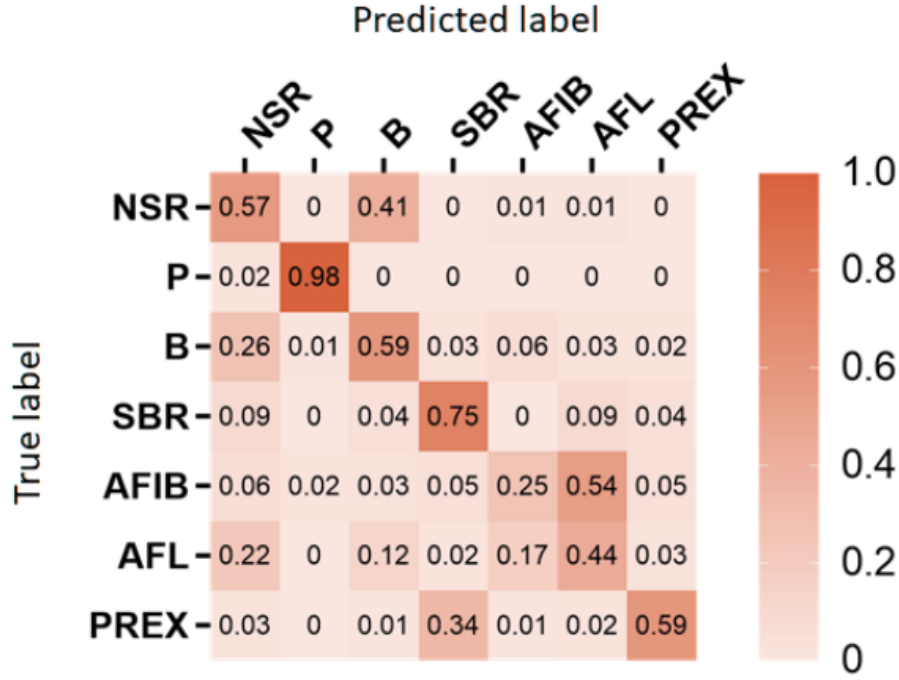


Figure 8.42: Normalized confusion matrix of the dataset used for the testing phase, with classes of Normal Sinus Rhythm (NSR), Paced Rhythm (P), Ventricular Bigeminy (B), Sinus Bradycardia (SBR), Atrial Fibrillation (AFIB), Atrial Flutter (AFL), and Wolff-Parkinson-White Syndrome (PREX).

experiments. Options for this improvement are the increase of the window size, which would help the algorithm to learn the differences of AFL and AFIB that, between other aspects, change the distance between peaks and extending the period, including more cycles, would make these easier to perceive. In future work, this algorithm will be tested with a more powerful hardware setting to be able to encompass this parameter change.

8.4.3.4 Comparative Study

The work of Hannun et al. (2019) [119], which is a leading article on this field published in Nature Medicine journal, was capable of reaching an average sensitivity of 75.2% for a 12 class model, compared to the average sensitivity of 61.1% for the proposed model. However, it is important to note that in the mentioned study, AFIB and AFL classes were combined into one unique class, due to their morphological similarities [119]. To make a comparative study we merged these two classes resulting in the confusion matrix depicted in Figure 8.43, which displays a significant change compared to the previous results (Fig. 8.42). Table 8.8 shows the numerical analysis for this change, reflecting an

Table 8.7: classification performance (in %) for the arrhythmia detection model

Class	Sensitivity	Specificity
NSR	63.3	85.5
P	97.1	99.7
B	36.3	93.7
SBR	62.6	96.3
AFIB	49.1	90.1
AFL	38.1	91.9
PREX	81.3	94.5

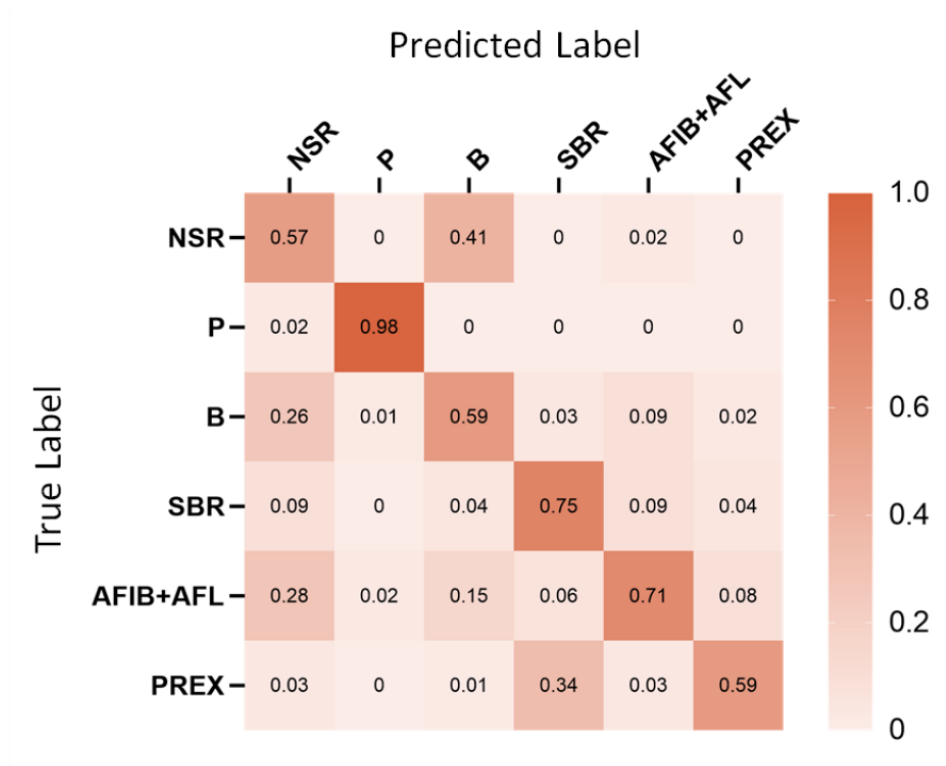


Figure 8.43: Normalized confusion matrix of the dataset used for the testing phase, with classes Normal Sinus Rhythm (NSR), Paced Rhythm (P), Ventricular Bigeminy (B), Sinus Bradycardia (SBR), Atrial Fibrillation with Atrial Flutter (AFIB + AFL), and Wolff-Parkinson-White Syndrome (PREX).

increase of the specificity to 84.8%, but decreasing the average sensitivity of the model to 70.9%, a value that only differs from the mentioned study by 4.3%. Note that these results are a consequence of the mismatch given by the B class, which displays a very low sensitivity value, as it is confused with a normal signal. This table also presents the comparative study with the overlapping classes with Hannun et al. (2019) [119].

Table 8.8: Classification performance (in %) for the arrhythmia detection model with merged AFIB and AFL and comparison with Hannun et al. (2019) [119]

Class	Sens	Sens of [119]	Spec	Spec of [119]
NSR	63.4	95.0	85.5	85.9
P	97.1	—	99,7	—
B	36.3	92.1	93,7	99.6
SBR	96,3	—	96,3	—
AFIB+AFL	84.8	86.1	90,9	94.1
PREX	81.3	65.1	94,5	98.6

Even though this algorithm does not outperform these state-of-the-art results, it empowers the view that the improvement of this architecture could provide outstanding results in the future. Even though some classes display a low accuracy when compared to the state of the art, the sensitivity is very close to the compared study. Bear in mind that this algorithm solely made to classify the eleven different classes of arrhythmia events, while our approach has the purpose of learning the general morphology of the healthy ECG detecting it from a modified version.

Conclusion and Future Remarks

By analyzing and presenting several architectures and after proving their usefulness in the research of patterns within the biosignal field, this thesis has shown that Deep Neural Networks (DNN) has a huge potential in providing new solutions and to the field. Some mechanisms of these architectures may be exploited for unveiling how biosignals work and the reason behind the classification of some pathological events.

This chapter has been segmented in distinct subtopics, at first the general conclusions for each application scenario will be presented, followed by the presentation of the overall contributions and outcomes. A summary of how the Explainable Artificial Intelligence (AI) can be used in these types of architectures and finally a personal view on how this area can be improved and what the future may hold regarding AI applied to physiologic signals.

9.1 Application Scenarios

9.1.1 Biosignal Synthesis

Starting with the synthesis study we can say that the achievements of this work were the ability to replicate the morphology of the three presented biosignals using one of the proposed architectures displayed in Chapter 7. One of the main aspects of this innovation that differs from previous literature is the capacity to learn and replicate several signals with no given human-extracted features.

The reconstruction of physiological signals from heavily noisy data with the use of these models is also one of the possible applications for this study. The application of the quantization step with higher detail, using a higher number of dimensions and applying smoothing to the quantized signal, and the incorporation of parameters that could influence the morphology of the wave, new realistic signals could be generated. These synthesized

signals could augment the data for training other Machine Learning (ML) algorithms. Higher realism and quality of the signal could be achieved by incorporating a discriminator during the training of the model, such as the ones used in Generative Adversarial Networks (GAN) architectures. This enhancement would add a new dimension in the perception of the morphological and spectral aspects of the signal.

Further observation of the learned models can be directed to the search of the internal neural structures that generate the morphological aspects of the signal. Even though the biosignal is a representation of the physiological aspects of the human body, the real mechanisms of the physical expression are concealed by external influences and the sequential patterns are not fully understood due to perceptual randomness. Therefore, the exploration of the inner workings that the model undergoes during the learning and synthesis of the biosignals' morphological characteristics could give hints about the original signal's generation. This information may give valuable insights on how to deliver novel procedures for decision making, supporting medical experts.

By trying to predict a signal (A) with a model that trained with that same signal will produce a lower error than when that same model is trying to predict a different signal (B). By analyzing the difference between both morphologies (the difference between A and B) it is possible to infer about aspects that differentiate them, for example, the type of signal, the sensor placement, or even the person who generated them. Likewise, applying this algorithm using segments of normal versus pathological physiological signals, the shown deviation from the trained model could also report the presence of symptomatic events. Following this reasoning, one of the applications made to this model was the implementation regarding biometric studies, as the model that learned a subject's signal was able to differentiate from another subject's ECG.

9.1.2 Biometric Systems

Regarding the spectrogram analysis for biometric systems, we can conclude that through the recognition of the temporal and spectral information we were able to confidently attribute small segments of ECG signals to the subject from whom they were collected.

One of the advantages of this algorithm is its robustness to variations in the moment of signal acquisition, as spectrograms with small offsets in time were correctly identified without the need for QRS detection.

Although the achieved results showed good performance, there is still a considerable margin for improvement. The use of transfer learning in this set could provide an internal model, closer to the random distribution that it currently has when the training is started. By representing the latent representation of the signal, especially in databases with a low amount of data, such as is the case of the ECG-ID database, could improve the results significantly, thus allowing the system to recognize differences in spectrograms with less training samples. Further development could also be made regarding the signal acquisition and the preprocessing step, as the effect of noise and artifacts present in the signal is a

probable cause for recognition error.

While this example demonstrates there is potential in the use of spectrograms for ECG biometric recognition applications, future work should include testing in data acquired in different periods, separated by several months or years, to prove the reliability of these systems over time. The application of DNN only using the signal morphology, without the use of human-engineered features was also deemed successful in both identification and authentication paradigms.

In the case of using the signal morphological aspects, the study has also shown the usefulness of DNN in enhancing current biometric systems. These technologies allied with the massive collection and storage of biological data will provide powerful and robust systems, increasing the level of security, reducing the concerns of counterfeiting actions.

Even though the Recurrent Neural Networks (RNN) and Temporal Convolutional Neural Network (TCNN) systems provided good results, their limitations could be surpassed if both architectures were combined. Other state-of-the-art downsampling methods could also be applied to mitigate the loss of information while increasing the training speed. The noise contamination in the ECG samples is also a major concern, by employing other filtering methods the system could learn better the intricate characteristics, only present in a clean signal.

The next steps to implement in a real-life scenario would be the introduction of transfer learning in the training process, the acquisition of more time data per person, the introduction of more individuals, and the increase of the number of acquisitions through time. The study of the impact of the noise and the scalability of these systems would give an added value.

One factor that could give a higher generalization and detection capabilities would be the inclusion of different electrode configurations for the clinical-grade ECG so that the DNN model learns the different angles of the signal gaining robustness over the different placements. The analysis in the context of different emotional and mental states could also be very important, as the ECG signal can change under different conditions.

9.1.3 Abnormally detection

As for the last example of the application of DNN architectures, the objective was to develop a framework that detects abnormalities in the normal pattern of the ECG signal, by creating a model that learns the key characteristics of a normal ECG cycle. The resultant models were capable of detecting two types of ECG events: noise and pathological events.

A machine-learned extractor was developed to be included in the architectures for noise and arrhythmia detection, by using the encoder module of a trained autoencoder. From the shown results it is suggested that by successfully reproducing the input signals, the autoencoder learns the key characteristics of a normal ECG cycle.

For the noise detection models, reaching state-of-the-art performance in the case of binary classification, despite achieving lower results in discriminating different types of

noise. In the case of the arrhythmia decision system, the model did not perform as well as the compared state-of-the-art accuracy, but the sensitivity and specificity showed to be promising.

The improvement of the system could rely on the increase of the power for the encoder by supplying it with more ECGs. As the autoencoder learns and replicates an increasing quantity of different segments of asymptomatic signals, the more this algorithm will embrace the abstract notion of what is an ECG. The presence of more noisy and symptomatic episodes would also contribute to the classification blocks since the dataset was limited in the number of provided segments.

In sum, the performance of these algorithms proves that it is possible to detect generic abnormal events in ECG using the proposed architectures. Therefore, the implementation of an encoder for classification algorithms proves to be effective in the detection of abnormal events by learning the signals' key characteristics.

As for the detecting of each type of abnormal event the model did not reach the state-of-the-art results as expected since the algorithms developed sacrifice accuracy to reach better generalizations for all ECG signals to be applicable in real life. Applications that understand the deviation from the normal signal into unseen pathologies and the possibility of creating new classes that did not exist before, could be extremely beneficial for the future not only for ECG but also other physiological signals.

One example is the application of this framework to active learning mechanisms as most of the available algorithms are conditioned to specific conditions, while not detecting others. The classifier's ability to detect similar abnormalities in the ECG signal could actively create new labels for patterns that diverge from the normal signal while increasing new diagnostic possibilities. That said, this technique could not only aid in the diagnostic of arrhythmias, but also contribute to the study of these pathological anomalies, by finding correlations between the different expressions of the ECG morphology, and also by exploring these manifestations with different variables in mind such as the patient's gender, age, medications, among other factors. Another example is the detection of intervals of interest from hours of patient data collected from wearable devices, the medical expert could focus only on the parts which are deviated from the asymptomatic signal. This would help a smoother and faster diagnostic while having a more informed decision.

This method helps to increase the possibilities, not only for detection but also for studying what exactly is a normal cycle and which are the deviation patterns. By having early detection procedures, one could early seek for medical help without developing a symptomatic episode. This preventive point of view could represent a major change in perspective on how medical care should be delivered in the future.

9.2 Contributions

The outcomes of this thesis are mostly applied to the area of machine learning and biomedical engineering. Some of them were applied in other research projects and institutions

proving the potential of the application of these applications.

- Biosignal processing:
 - Signal preprocessing methods optimized to feed normalized signals into Neural Networks (NN), such as:
 - * Normalization using a time window;
 - * Quantization with the clipped edges to reduce artifacts interference;
 - * Rules regarding window segmentation for an optimized learning process.
- Feature extraction:
 - Using Autoencoders to extract features and how to combine them with a classification module for joint action of generation and decision-making.
- Generation:
 - Using RNN to generate ECG, Electromyogram (EMG) and Respiration Signal (RESP) resorting to a quantization process;
 - Generation of biosignals using an agnostic approach, being possible to synthesize other types based on a related one;
 - The development of ECG Generator of Representative Encoding of Style and Symptoms (EGRESS) for the generation of an ECG with a mixture of characteristics of two different ECG, using the traits of each one, producing a signal that does not exist in the real world.
- Classification Systems:
 - Regarding the biometric system, the use of the prediction error of a RNN as a measurement of similarity between ECG;
 - The creation Relative Score Threshold Classifier (RSTC) which normalizes and compares the similarities between time-windows;
- Explainable AI:
 - Showing how the states and gates of the RNN architecture and the Convolution Neural Networks (CNN) filter can contribute to understanding the inner mechanisms of the networks;
 - Showing how the learning process occurs in the ECG segments by generating these signals in different epochs of the training process.
- Machine Learning:
 - The development of EGRESS that creates a mixture of two 1D of characteristics of two different vectors producing a non-existent class.

9.2.1 List of publications

9.2.1.1 Journal Papers

The following list comprises the list of publications published during this thesis. They are the output collaborating with projects Augmented Human Assistance (AHA) and Intelligent Networked Robot Systems for Symbiotic Interaction with Children with Impaired Development (INSIDE) and of the collaboration of the institutes of Fraunhofer AICOS, PLUX, and Ural Federal University.

- Biosignals learning and synthesis using deep neural networks.
D. Belo, J. Rodrigues, J.R. Vaz, P. Pazarat-Correia, H. Gamboa
BioMedical Engineering OnLine 16 (1), 115 7. 2017.
- Noise detection on ECG based on agglomerative clustering of morphological features.
J. Rodrigues, D. Belo, H. Gamboa
Computers in biology and medicine 87, 322-334. 2017.
- Comparison of machine learning methods for the arterial hypertension diagnostics.
V.S. Kublanov, A.Y. Dolganov, D. Belo, H. Gamboa
Applied bionics and biomechanics. 2017.
- Project INSIDE: towards autonomous semi-unstructured human-robot social interaction in autism therapy.
F.S. Melo, A. Sardinha, D. Belo, M. Couto, M. Faria, A. Farias, H. Gamboa, C. Jesus, M. Kinarullathil, P. Lima, L. Luz, A. Mateus, I. Melo, P. Moreno, D. Osório, A. Paiva, J. Pimentel, J. Rodrigues, P. Sequeira, R. Solera-Ureña, M. Vasco, M. Veloso, R. Ventura.
Artificial intelligence in medicine 96, 198-216 5. 2019.
- SSTs: A syntactic tool for pattern search on time series.
J. Rodrigues, D. Folgado, D. Belo, H. Gamboa.
Information Processing & Management 56 (1), 61-76. 2019.
- ECG Biometrics Using Deep Learning and Relative Score Threshold Classification
D. Belo, N. Bento, H. Silva, A. Fred, H. Gamboa
Sensors 20 (15), 4078, 2020.

9.2.1.2 List of conference articles

- Development of the decision support system in the treatment of arterial hypertension application of artificial neural networks for evaluation of heart rate variability signals.
A. Dolganov, V. Kublanov, D. Belo, H. Gamboa.
11th International Conference on Bio-Inspired Systems and Signal Processing. 2018.

- ECG Biometrics Using Spectrograms and Deep Neural Networks.
N. Bento, D. Belo, H. Gamboa
International Conference on Biomedical Imaging, Signal Processing 2018. 2018.
- Detection of abnormalities in Electrocardiogram (ECG) using Deep Learning.
J. Pestana, D. Belo, H. Gamboa.
13th International Conference on Bio-inspired Systems and Signal Processing 2019.
2019.

9.3 Future Remarks

The current state of AI is impressive and this technology has a fast-pace evolution. This moment is only the beginning, and with a higher understanding of how these systems work and how they can be matched together, machines with multimodal and continuous learning capabilities will emerge.

The evolution of these technologies will depend on, but not limited to, two different architectural considerations: the fusion between several DNN architectures, by using multimodal network approaches, and the need to explain the decision process of AI algorithms. This section provides the author's views on how the field of Deep Learning (DL) can improve in both these approaches and also the future on AI applied to the healthcare field.

9.3.1 Fusion between networks

The multimodal approach presented in chapter 7 on subsection 7.1.1 describes how it is possible to join different networks using fusion layers, but in the future, this can be improved even further with a hierarchical approach to join different architectures specialized in various biosignals domains into one single network.

By grouping different related channels with a sum or average fusion later and joining the results of all groups with concatenation will join the information of different related domains with a less computational cost. For example, by organizing different Electroencephalogram (EEG) channels by area of the brain cortex, and training a network per channel, it is possible to fuse the latent vectors by average each channel which belongs to a specific area of the brain, and, in the end, join each latent vector by a concatenation rule and attach it to a classification module. New patterns could be detected for different brain area activations.

Another example is to have each channel provide information about the different brain waves and treat a channel as a group. When joining together all channels it will provide information about sleep stages, or symptomatic events that are related to brainwave disturbance. This hierarchical approach can also be used in scaling mechanisms, for example, looking into a time window of 128 samples of an ECG and segment it into one window of 256, two windows of 128 and four windows provide 3 different aspects of the time-window. When making three scaling groups (i.e. 256 as group 1, two 128 sampled

windows in group 2 and the rest of the windows in group 3) and concatenate the fusion layers, giving hints about the segment of the time-window that most contribute to the symptom that was detected.

Nowadays the saying of "no free lunch" applied to these technologies, which reflects the limitations that the models have in performing well in a specific task, will be obsolete due to the combination of different specialized networks. It is possible to compare the fusion between different networks with the interconnection of the different specialized brain areas, or even the collective made of different people. Teams with people with different knowledge areas can bring together outstanding solutions when cooperating.

9.3.2 Explainable Artificial Intelligence

Explainable AI (XAI) systems are methods placed on top of models that could provide information about the results in a way human experts can understand, as opposed to the concept of "black box". This concept may be aligned with a trade-off between performance and the understanding of the decisions made by DNN models.

In the case of biosignal DNN, the information could be shared in terms of the created filters and features. Theoretically, one could have access to the filters created during the training of the networks and visualize what segments were activated during the classification method. In the case of Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) based architectures, one can look into the output of the network and specific hidden nodes states. By studying the mechanics of how the gates work and the activation of each in different aspects of the signal may inform about the differences between the classified as symptomatic and asymptomatic time-window, for example.

One method that could also be implemented is the study of extracted features using clustering methods in models with autoencoders. By using the configurations in which the encoder, decoder, and classifier are trained at the same time, one can explore the different features produced. For different labeled windows, the features produced will be different, and therefore, by generating a signal and comparing it with the input signal it can be possible to check if the network is "seeing" what it is supposed to. At the same time as exploring different windows and clustering the extracted features, one could see the differences in the recreations from the different clusters. One other option would be to feature engineer and explore how the different features have an effect in the generation of the signal, and therefore how it has an impact on the classification method. E.g. supplying different values to one of the positions of the latent vector and compare the generation of the signal with the input signal it is supposed to reproduce, and check the classification made.

At last in combination with experts, one could make a combined model with different scales of a signal specifically for an issue. For the sake of simplicity, the symptomatic events of an ECG during Atrial Fibrillation will be used as an example. As one segments a time-window containing two heart-beats into different regions of the wave, regions which

must have a specific and different morphology from the normal signal when there is a symptomatic episode. As the windows are segmented into smaller windows containing these regions of the wave and training it with its autoencoder together and fuse all the latent vectors, the least dense layers will then classify if the signal has Atrial Fibrillation or not. By exploring the different regions of the signal and the different created latent spaces, one can see which segments activated the classification and therefore understand some of the reasoning behind the decision of the network.

9.3.3 AI in healthcare

In the healthcare field, learning from multiple sources, such as patient demographics, drug information, biosignals, and chemical analysis could make new systems that work together to help the doctor to find a diagnostic. If different systems are trained separately and an interface could merge them and by explaining each decision, it would provide more informed outcomes to help the medical doctor diagnostic. In the future doctors and AI will work together in finding a solution with an interface where a dialog will be made between a therapist and the machine, providing several solutions and possible diagnostics. In the future, AI models will be continuously trained and personalized to the specific physician.

Another interesting emergent technology is the Generative Adversary Networks, which could be used as the mechanism we use when dreaming. During our young ages, we simulate our environment and make scenarios to exercise the brain and form new pathways while sleeping [212]. Analogously, GAN can work as an off-line trainer, which would improve performance to be applied in real-life.

This is an exciting time to live, with the increase of power of Graphics Processing Unit (GPU) and specific for NN computation, the future will hold systems that our imagination cannot conceive. In a less positive note politics will make laws to restrict some AI practices and data will be more expensive to get. But in a few decades, we will be dependant on these technologies, as they will prove to make life easier, removing most of the tedious repetitive tasks. And in the healthcare industry, it will help us to provide the next generation of medicine, helping us to detect patterns not seen before and provide new solutions that are not bound by the common human thinking.

Bibliography

- [1] *Retrato da Saúde*. Ministério da Saúde, 2018. URL: https://www.sns.gov.pt/wp-content/uploads/2018/04/RETRATO-DA-SAUDE_2018_compressed.pdf.
- [2] D. Belo et al. “Quality of service in transplantation via the electronic medical record”. In: *Proceedings of the 7th Annual Industrial Simulation Conference, Loughborough, UK, 2009*. EUROSIS-ETI. 2009, pp. 1–6.
- [3] B. Chaudhry et al. “Systematic review: impact of health information technology on quality, efficiency, and costs of medical care”. In: *Annals of internal medicine* 144.10 (2006), pp. 742–752.
- [4] *World Industry Outlook, Healthcare and Pharmaceuticals*. 2018.
- [5] F. F. M. dos Santos. *PORDATA, Bases de Dadas Portugal Contemporâneo - Saúde*. URL: <https://www.pordata.pt/Tema/Europa/Saude-26>.
- [6] Deloitte. *The new physics of financial services / How artificial intelligence is transforming the financial ecosystem*. 2018. URL: <https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Financial-Services/gx-fsi-ai-wef-summary.pdf>.
- [7] *OECD Health Statistics*. 2018. URL: <https://doi.org/10.1787/health-data-en>.
- [8] *Innovation in Medicine 2018: From surviving to thriving: culture in healthcare*. 2018. URL: <https://www.rcplondon.ac.uk/news/innovation-medicine-2018-surviving-thriving-culture-healthcare>.
- [9] E. Mahoney. *Q&A: NASA’s New Spaceship*. 2018. URL: <https://www.nasa.gov/feature/questions-nasas-new-spaceship>.
- [10] A. Roda et al. “Advanced biosensors for monitoring astronauts’ health during long-duration space missions”. In: *Biosensors and Bioelectronics* 111 (2018), pp. 18–26.
- [11] J. Rodrigues, D. Belo, and H. Gamboa. “Noise detection on ECG based on agglomerative clustering of morphological features”. In: *Computers in biology and medicine* 87 (2017), pp. 322–334.
- [12] L. Deng. “Artificial Intelligence in the Rising Wave of Deep Learning: The Historical Path and Future Outlook [Perspectives]”. In: *IEEE Signal Processing Magazine* 35.1 (2018), pp. 180–177. ISSN: 1053-5888. DOI: 10.1109/MSP.2017.2762725.

- [13] K. Lewandowski. *Survey Finds Machine Learning and Artificial Intelligence are Top Business Priorities*. 2018. URL: <https://www.memsql.com/releases/oreilly-survey/>.
- [14] E. Kaniusas. *Biomedical signals and sensors I: Linking physiological phenomena and biosignals*. Springer Science & Business Media, 2012.
- [15] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. “Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription”. In: *ArXiv e-prints* (June 2012). arXiv: 1206.6392 [cs.LG].
- [16] *Artificial Intelligence Market*. URL: <https://www.marketsandmarkets.com/Market-Reports/artificial-intelligence-market-74851580.html>.
- [17] J. R. Hughes. “Gamma, fast, and ultrafast waves of the brain: their relationships with epilepsy and behavior”. In: *Epilepsy & Behavior* 13.1 (2008), pp. 25–31.
- [18] S. Alaybeyi et al. “Debunking Myths and Misconceptions About Artificial Intelligence”. In: *Gartner* (2019). URL: <https://www.gartner.com/doc/3898567?ref=mrktg-srch>.
- [19] Factank. “Nearly half of Americans use digital voice assistants, mostly on their smartphones”. In: *Pew Research Center* (2017). URL: <http://pewrsr.ch/2kquZ8H>.
- [20] A. Rao and M. Lieberman. *Bot.me: How artificial intelligence is pushing man and machine closer together*. 2017. URL: https://www.pwc.com/CISAI?WT.mc_id=CT1-PL52-DM2-TR1-LS4-ND6-BPA1-CN_CIS-AI-AIsocial.
- [21] *Uses of virtual assistants in the U.S. 2017 / Statistic*. 2017. URL: <https://www.statista.com/statistics/702926/united-states-digital-voice-assistants-survey-usage/>.
- [22] S. Ransbotham et al. *Artificial Intelligence in Business Gets Real*. 2018. URL: <https://sloanreview.mit.edu/projects/artificial-intelligence-in-business-gets-real/>.
- [23] Y. Bengio, P. Simard, and P. Frasconi. “Learning long-term dependencies with gradient descent is difficult”. In: vol. 5. 2. 1994, pp. 157–166.
- [24] E. John and A. C. Guyton. *Guyton and Hall Textbook of Medical Physiology*. Saunders/Elsevier, 2011.
- [25] A. Cohen. “Biomedical signals: Origin and dynamic characteristics; frequency-domain analysis”. In: *The biomedical engineering handbook* 2 (2000), pp. 951–74.
- [26] B. Malhotra. *Biosensors : fundamentals and applications*. Shawbury, United Kingdom: Smithers Rapra Technology, Ltd, 2017. ISBN: 978-1-91024-278-0.

-
- [27] H. Prance. “Sensor Developments for Electrophysiological Monitoring in Health-care”. In: *Applied Biomedical Engineering*. Ed. by G. D. Gargiulo and A. McEwan. Rijeka: IntechOpen, 2011. Chap. 12. DOI: 10.5772/22958. URL: <https://doi.org/10.5772/22958>.
 - [28] S. M. Borisov and O. S. Wolfbeis. “Optical biosensors”. In: *Chemical reviews* 108.2 (2008), pp. 423–461.
 - [29] C. McDonagh, C. S. Burke, and B. D. MacCraith. “Optical chemical sensors”. In: *Chemical reviews* 108.2 (2008), pp. 400–422.
 - [30] *Structure of the plasma membrane*. URL: <https://www.khanacademy.org/science/high-school-biology/hs-cells/hs-the-cell-membrane/a/structure-of-the-plasma-membrane>.
 - [31] M. Zamir and E. Ritman. *The physics of pulsatile flow*. Springer, 2000.
 - [32] N. Peña Pérez. “Windkessel modeling of the human arterial system”. B.S. thesis. 2016.
 - [33] H. Gamboa. “Multi-Modal Behavioral Biometrics Based on HCI and Electrophysiology”. PhD thesis. PhD thesis, Universidade Técnica de Lisboa, Instituto Superior Técnico, 2008.
 - [34] V. S. Kublanov et al. “Comparison of machine learning methods for the arterial hypertension diagnostics”. In: *Applied bionics and biomechanics* 2017 (2017).
 - [35] N. A. Shirwany and M.-h. Zou. “Arterial stiffness: a brief review”. In: *Acta Pharmacologica Sinica* 31.10 (2010), p. 1267.
 - [36] *Arterial tortuosity syndrome - Genetics Home Reference - NIH*. 2019. URL: <https://ghr.nlm.nih.gov/primer/illustrations/therapyvector>.
 - [37] P. Konrad. “The abc of emg”. In: *A practical introduction to kinesiological electromyography* 1.2005 (2005), pp. 30–35.
 - [38] S. M. L. V. P. Santos. “Explaining the Ergonomic Assessment of Human Movement in Industrial Contexts”. PhD thesis. 2019.
 - [39] A. Mallik, B. Bhowmick, and S. Alam. “A multi-sensor information fusion approach for efficient 3D reconstruction in smart phone”. In: *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*. The Steering Committee of The World Congress in Computer Science, Computer ... 2015, p. 291.
 - [40] W Kahle and M Frotscher. *Color Atlas of Human Anatomy Nervous System and Sensory Organs, Vol. 3*. 2003.
 - [41] D. Purves et al. *Neuroscience*. Sinauer Associates, Inc., 2012.
 - [42] *The Upper Motor Neurons (Motor Systems) Part 1*. URL: <http://what-when-how.com/neuroscience/the-upper-motor-neurons-motor-systems-part-1/>.

- [43] E. Musk et al. “An integrated brain-machine interface platform with thousands of channels”. In: *BioRxiv* (2019), p. 703801.
- [44] J. J. Braithwaite et al. “A guide for analysing electrodermal activity (EDA) & skin conductance responses (SCRs) for psychological experiments”. In: *Psychophysiology* 49.1 (2013), pp. 1017–1034.
- [45] M. Niemann, A. Prange, and D. Sonntag. “Towards a multimodal multisensory cognitive assessment framework”. In: *2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE. 2018, pp. 24–29.
- [46] *All About EDA Part 1: Introduction to Electrodermal Activity*. 2017. URL: <https://support.mindwaretech.com/2017/12/all-about-eda-part-1-introduction-to-electrodermal-activity/>.
- [47] B. J. Copeland and D. Proudfoot. “Alan Turing’s forgotten ideas in computer science”. In: *Scientific American* 280.4 (1999), pp. 98–103.
- [48] H. Sak, A. Senior, and F. Beaufays. “Long short-term memory recurrent neural network architectures for large scale acoustic modeling”. In: *Fifteenth annual conference of the international speech communication association*. 2014.
- [49] Y. LeCun et al. “Backpropagation applied to handwritten zip code recognition”. In: *Neural computation* 1.4 (1989), pp. 541–551.
- [50] D. Baillot. *Why are neuron axons long and spindly? Study shows they’re optimizing signaling efficiency*. 2018. URL: <https://medicalxpress.com/news/2018-07-neuron-axons-spindly-theyre-optimizing.html>.
- [51] Šarūnas Raudys. “Evolution and generalization of a single neurone: I. Single-layer perceptron as seven statistical classifiers”. In: *Neural Networks* 11.2 (1998), pp. 283–296. ISSN: 0893-6080. DOI: [https://doi.org/10.1016/S0893-6080\(97\)00135-4](https://doi.org/10.1016/S0893-6080(97)00135-4). URL: <http://www.sciencedirect.com/science/article/pii/S0893608097001354>.
- [52] I. Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.
- [53] S. Saha. *A Comprehensive Guide to Convolutional Neural Networks - the ELI5 way*. 2018. URL: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
- [54] R. Jozefowicz, W. Zaremba, and I. Sutskever. “An empirical exploration of recurrent network architectures”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Lille, France, 2015.
- [55] Y. Bengio et al. “Greedy Layer-Wise Training of Deep Networks”. In: *Advances in Neural Information Processing Systems 19*. Ed. by B. Schölkopf, J. C. Platt, and T. Hoffman. MIT Press, 2007, pp. 153–160.

-
- [56] I. Sutskever, J. Martens, and G. E. Hinton. “Generating Text with Recurrent Neural Networks.” In: *ICML*. Ed. by L. Getoor and T. Scheffer. Omnipress, 2011, pp. 1017–1024. URL: <http://dblp.uni-trier.de/db/conf/icml/icml2011.html#SutskeverMH11>.
- [57] S. Hochreiter and J. Schmidhuber. “Long Short-Term Memory”. In: *Long Short-Term Memory*. Vol. 9. 1997, pp. 1735–1780.
- [58] A. Graves and J. Schmidhuber. “Framewise phoneme classification with bidirectional LSTM and other neural network architectures”. In: *Neural Networks* 18 (2005), pp. 602–610. DOI: 10.1016/j.neunet.2005.06.042. URL: <http://dx.doi.org/10.1016/j.neunet.2005.06.042>.
- [59] Z. Wu and S. King. “Investigating gated recurrent neural networks for speech synthesis”. In: *CoRR* abs/1601.02539 (2016). URL: <http://arxiv.org/abs/1601.02539>.
- [60] K. Greff et al. “LSTM: A Search Space Odyssey”. In: *CoRR* abs/1503.04069 (2015). URL: <http://arxiv.org/abs/1503.04069>.
- [61] S. Kaae Sønderby and O. Winther. “Protein Secondary Structure Prediction with Long Short Term Memory Networks”. In: *ArXiv e-prints* (Dec. 2014). arXiv: 1412.7828 [q-bio.QM].
- [62] H. Sak, A. Senior, and F. Beaufays. “Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling”. In: *Annual Conference of the International Speech Communication Association - Interspeech 2014*. Singapore, 2014, pp. 338–342.
- [63] D. Eck and J. Schmidhuber. “A first look at music composition using lstm recurrent neural networks”. In: *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale* 103 (2002).
- [64] H. Zen and H. Sak. “Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis”. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*. 2015, pp. 4470–4474. DOI: 10.1109/ICASSP.2015.7178816. URL: <http://dx.doi.org/10.1109/ICASSP.2015.7178816>.
- [65] A. Graves, A. Mohamed, and G. E. Hinton. “Speech Recognition with Deep Recurrent Neural Networks”. In: *CoRR* abs/1303.5778 (2013). URL: <http://arxiv.org/abs/1303.5778>.
- [66] E. Marchi et al. “Multi-resolution linear prediction based features for audio onset detection with bidirectional LSTM neural networks”. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 2164–2168.

- [67] A. Graves et al. “A novel connectionist system for unconstrained handwriting recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 31.5 (2009), pp. 855–868.
- [68] V. Pham, C. Kermorvant, and J. Louradour. “Dropout improves Recurrent Neural Networks for Handwriting Recognition”. In: *CoRR* abs/1312.4569 (2013). URL: <http://arxiv.org/abs/1312.4569>.
- [69] M.-T. Luong et al. “Addressing the Rare Word Problem in Neural Machine Translation”. In: (2015). URL: <http://arxiv.org/abs/1410.8206v4>.
- [70] J. Donahue et al. “Long-term recurrent convolutional networks for visual recognition and description”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 2625–2634.
- [71] O. Vinyals et al. “Show and Tell: A Neural Image Caption Generator”. In: *CoRR* abs/1411.4555 (2014). URL: <http://arxiv.org/abs/1411.4555>.
- [72] M. Ren, R. Kiros, and R. S. Zemel. “Image Question Answering: A Visual Semantic Embedding Model and a New Dataset”. In: *CoRR* abs/1505.02074 (2015). URL: <http://arxiv.org/abs/1505.02074>.
- [73] X. Guo et al. “Deep clustering with convolutional autoencoders”. In: *International conference on neural information processing*. Springer. 2017, pp. 373–382.
- [74] L. Gondara. “Medical image denoising using convolutional denoising autoencoders”. In: *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2016, pp. 241–246.
- [75] L. Theis et al. “Lossy image compression with compressive autoencoders”. In: *arXiv preprint arXiv:1703.00395* (2017).
- [76] Y. Burda, R. Grosse, and R. Salakhutdinov. “Importance weighted autoencoders”. In: *arXiv preprint arXiv:1509.00519* (2015).
- [77] D. P. Kingma and M. Welling. “Stochastic gradient VB and the variational auto-encoder”. In: *Second International Conference on Learning Representations, ICLR*. Vol. 19. 2014.
- [78] C.-S. Lee et al. “Human vs. computer go: Review and prospect [discussion forum]”. In: *IEEE Computational intelligence magazine* 11.3 (2016), pp. 67–72.
- [79] T. Karras, S. Laine, and T. Aila. “A Style-Based Generator Architecture for Generative Adversarial Networks”. In: *CoRR* abs/1812.04948 (2018). arXiv: 1812.04948. URL: <http://arxiv.org/abs/1812.04948>.
- [80] M.-H. Maras and A. Alexandrou. “Determining authenticity of video evidence in the age of artificial intelligence and in the wake of Deepfake videos”. In: *The International Journal of Evidence & Proof* 23.3 (2019), pp. 255–262.

-
- [81] W Schiffmann, M Joost, and R Werner. “Optimization of the backpropagation algorithm for training multilayer perceptrons”. In: *University of Koblenz: Institute of Physics* (1994).
 - [82] R. J. Williams and J. Peng. “An efficient gradient-based algorithm for on-line training of recurrent network trajectories”. In: *Neural computation* 2.4 (1990), pp. 490–501.
 - [83] M. Stewart. 2019. URL: <https://towardsdatascience.com/>.
 - [84] S. K. Kumar. “On weight initialization in deep neural networks”. In: *arXiv preprint* (2017).
 - [85] K. He et al. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.
 - [86] L. Bottou. “Large-scale machine learning with stochastic gradient descent”. In: *Proceedings of COMPSTAT-2010*. Springer, 2010, pp. 177–186.
 - [87] N. Qian. “On the momentum term in gradient descent learning algorithms”. In: *Neural networks* 12.1 (1999), pp. 145–151.
 - [88] Y. Nesterov. “A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$ ”. In: 1983.
 - [89] J. Duchi, E. Hazan, and Y. Singer. “Adaptive subgradient methods for online learning and stochastic optimization”. In: *Journal of Machine Learning Research* 12.Jul (2011), pp. 2121–2159.
 - [90] M. D. Zeiler. “ADADELTA: an adaptive learning rate method”. In: *arXiv preprint* (2012).
 - [91] G. Hinton, N. Srivastava, and K. Swersky. “Neural networks for machine learning lecture 6a overview of mini-batch gradient descent”. In: ().
 - [92] D. P. Kingma and J. Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
 - [93] T. Tieleman and G. Hinton. “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude”. In: *COURSERA: Neural networks for machine learning* 4.2 (2012), pp. 26–31.
 - [94] R. Rodrigues and P. Couto. “A Neural Network Approach to ECG Denoising”. In: *CoRR* abs/1212.5217 (2012).
 - [95] H. P. Martinez, Y. Bengio, and G. N. Yannakakis. “Learning deep physiological models of affect”. In: *IEEE Computational Intelligence Magazine* 8.2 (2013), pp. 20–33.

- [96] X. Zhu et al. “EOG-based drowsiness detection using convolutional neural networks”. In: *2014 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2014, pp. 128–134.
- [97] W. Geng et al. “Gesture recognition by instantaneous surface EMG images”. In: *Scientific reports* 6 (2016), p. 36571.
- [98] U. Cote-Allard et al. “Transfer learning for sEMG hand gestures recognition using convolutional neural networks”. In: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2017, pp. 1663–1668.
- [99] O. Tsinalis et al. “Automatic sleep stage scoring with single-channel EEG using convolutional neural networks”. In: *arXiv preprint arXiv:1610.01683* (2016).
- [100] Y. Xia et al. “Detecting atrial fibrillation by deep convolutional neural networks”. In: *Computers in biology and medicine* 93 (2018), pp. 84–92.
- [101] S. P. Shashikumar et al. “A deep learning approach to monitoring and detecting atrial fibrillation using wearable technology”. In: *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE. 2017, pp. 141–144.
- [102] E. J. da Silva Luz et al. “Learning Deep Off-the-Person Heart Biometrics Representations”. In: *IEEE Transactions on Information Forensics and Security* 13.5 (2018), pp. 1258–1270. DOI: 10.1109/TIFS.2017.2784362.
- [103] J. P. Dominguez-Morales et al. “Deep neural networks for the recognition and classification of heart murmurs using neuromorphic auditory sensors”. In: *IEEE transactions on biomedical circuits and systems* 12.1 (2017), pp. 24–34.
- [104] T. Nilanon et al. “Normal/abnormal heart sound recordings classification using convolutional neural network”. In: *2016 Computing in Cardiology Conference (CinC)*. IEEE. 2016, pp. 585–588.
- [105] F. Pan et al. “Variation of the Korotkoff stethoscope sounds during blood pressure measurement: Analysis using a convolutional neural network”. In: *IEEE journal of biomedical and health informatics* 21.6 (2017), pp. 1593–1598.
- [106] P. Xia, J. Hu, and Y. Peng. “EMG-based estimation of limb movement using deep learning with recurrent convolutional neural networks”. In: *Artificial organs* 42.5 (2018), E67–E77.
- [107] A. Sengur et al. “Classification of amyotrophic lateral sclerosis disease based on convolutional neural network and reinforcement sample learning algorithm”. In: *Health information science and systems* 5.1 (2017), p. 9.
- [108] Q. Zhang, D. Zhou, and X. Zeng. “HeartID: A multiresolution convolutional neural network for ECG-based biometric human identification in smart health applications”. In: *Ieee Access* 5 (2017), pp. 11805–11816.

-
- [109] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.
 - [110] W. Chen et al. “A CHF detection method based on deep learning with RR intervals”. In: *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2017, pp. 3369–3372.
 - [111] V. Maknickas and A. Maknickas. “Recognition of normal–abnormal phonocardiographic signals using deep convolutional neural networks and mel-frequency spectral coefficients”. In: *Physiological measurement* 38.8 (2017), p. 1671.
 - [112] M. M. Al Rahhal et al. “Deep learning approach for active classification of electrocardiogram signals”. In: *Information Sciences* 345 (2016), pp. 340–354.
 - [113] Z. Yao, Z. Zhu, and Y. Chen. “Atrial fibrillation detection by multi-scale convolutional neural networks”. In: *2017 20th International Conference on Information Fusion (Fusion)*. IEEE. 2017, pp. 1–6.
 - [114] 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.
 - [115] F. yan Zhou, L. peng Jin, and J. Dong. “Premature ventricular contraction detection combining deep neural networks and rules inference”. In: *Artificial Intelligence in Medicine* 79 (2017), pp. 42 –51. ISSN: 0933-3657. DOI: <https://doi.org/10.1016/j.artmed.2017.06.004>. URL: <http://www.sciencedirect.com/science/article/pii/S0933365716305668>.
 - [116] W. Liu et al. “Real-Time Multilead Convolutional Neural Network for Myocardial Infarction Detection”. In: *IEEE Journal of Biomedical and Health Informatics* 22.5 (2018), pp. 1434–1444. DOI: 10.1109/JBHI.2017.2771768.
 - [117] U. R. Acharya et al. “Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals”. In: *Computers in Biology and Medicine* 100 (2018), pp. 270 –278. ISSN: 0010-4825. DOI: <https://doi.org/10.1016/j.compbimed.2017.09.017>. URL: <http://www.sciencedirect.com/science/article/pii/S0010482517303153>.
 - [118] S. Savalia and V. Emamian. “Cardiac Arrhythmia Classification by Multi-Layer Perceptron and Convolution Neural Networks”. In: *Bioengineering* 5.2 (2018), p. 35. ISSN: 2306-5354. DOI: 10.3390/bioengineering5020035. URL: <http://www.ncbi.nlm.nih.gov/pubmed/29734666><http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC6027502><http://www.mdpi.com/2306-5354/5/2/35>.

- [119] A. Y. Hannun et al. “Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network”. In: *Nature Medicine* 25.1 (2019), pp. 65–69. ISSN: 1078-8956. DOI: 10.1038/s41591-018-0268-3. URL: <http://www.ncbi.nlm.nih.gov/pubmed/30617320><http://www.nature.com/articles/s41591-018-0268-3>.
- [120] S. Ansari, J. Gryak, and K. Najarian. “Noise Detection in Electrocardiography Signal for Robust Heart Rate Variability Analysis: A Deep Learning Approach”. In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2018, pp. 5632–5635.
- [121] Q. Zhang et al. “Respiration-based emotion recognition with deep learning”. In: *Computers in Industry* 92 (2017), pp. 84–90.
- [122] H. Ryu, J. Park, and H. Shin. “Classification of heart sound recordings using convolution neural network”. In: *2016 Computing in Cardiology Conference (CinC)*. 2016, pp. 1153–1156.
- [123] M. Atzori, M. Cognolato, and H. Müller. “Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands”. In: *Frontiers in neurorobotics* 10 (2016), p. 9.
- [124] S. Hahm and J. Wang. “Silent speech recognition from articulatory movements using deep neural network.” In: *ICPhS*. 2015.
- [125] W.-L. Zheng and B.-L. Lu. “Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks”. In: *IEEE Transactions on Autonomous Mental Development* 7.3 (2015), pp. 162–175.
- [126] J. Yang et al. “Deep convolutional neural networks on multichannel time series for human activity recognition”. In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015.
- [127] R. K. Pathinarupothi et al. “Instantaneous heart rate as a robust feature for sleep apnea severity detection using deep learning”. In: *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE. 2017, pp. 293–296.
- [128] Y. Qiu, F. Xiao, and H. Shen. “Elimination of power line interference from ECG signals using recurrent neural networks”. In: *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2017, pp. 2296–2299.
- [129] S. Chen and Q. Jin. “Multi-modal Dimensional Emotion Recognition Using Recurrent Neural Networks”. In: *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*. AVEC ’15. Brisbane, Australia: ACM, 2015, pp. 49–56. ISBN: 978-1-4503-3743-4. DOI: 10.1145/2808196.2811638. URL: <http://doi.acm.org/10.1145/2808196.2811638>.

-
- [130] M. Thill et al. “Anomaly Detection in Electrocardiogram Readings with Stacked LSTM Networks”. In: (2019).
 - [131] R. S. Andersen, R. Peimankar, and S. Puthusserypady. “A Deep Learning Approach for Real-Time Detection of Atrial Fibrillation”. In: *Expert Systems with Applications* 115 (2018). DOI: 10.1016/j.eswa.2018.08.011.
 - [132] C. Thomae and A. Dominik. “Using deep gated RNN with a convolutional front end for end-to-end classification of heart sound”. In: *2016 Computing in Cardiology Conference (CinC)*. IEEE. 2016, pp. 625–628.
 - [133] C. Schölzel and A. Dominik. “Can electrocardiogram classification be applied to phonocardiogram data?—An analysis using recurrent neural networks”. In: *2016 Computing in Cardiology Conference (CinC)*. IEEE. 2016, pp. 581–584.
 - [134] A. Graves et al. “Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks”. In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML ’06. Pittsburgh, Pennsylvania, USA: ACM, 2006, pp. 369–376. ISBN: 1-59593-383-2. DOI: 10.1145/1143844.1143891. URL: <http://doi.acm.org/10.1145/1143844.1143891>.
 - [135] I. Sutskever, O. Vinyals, and Q. V. Le. “Sequence to Sequence Learning with Neural Networks”. In: *CoRR* abs/1409.3215 (2014). arXiv: 1409.3215. URL: <http://arxiv.org/abs/1409.3215>.
 - [136] H. Dong et al. “Mixed Neural Network Approach for Temporal Sleep Stage Classification”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 26.2 (2018), pp. 324–333. DOI: 10.1109/TNSRE.2017.2733220.
 - [137] T. J. Jun et al. “Premature ventricular contraction beat detection with deep neural networks”. In: *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2016, pp. 859–864.
 - [138] L. Diener, M. Janke, and T. Schultz. “Direct conversion from facial myoelectric signals to speech using deep neural networks”. In: *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2015, pp. 1–7.
 - [139] N. Ganapathy, R. Swaminathan, and T. M. Deserno. “Deep learning on 1-D biosignals: a taxonomy-based survey”. In: *Yearbook of medical informatics* 27.01 (2018), pp. 098–109.
 - [140] G. E. Hinton, S. Osindero, and Y. whye Teh. *A Fast Learning Algorithm for Deep Belief Nets*. 2006.
 - [141] V. Gandhi et al. “Evaluating Quantum Neural Network filtered motor imagery brain-computer interface using multiple classification techniques”. In: *Neurocomputing* 170 (2015). Advances on Biological Rhythmic Pattern Generation: Experiments, Algorithms and Applications Selected Papers from the 2013 International

- Conference on Intelligence Science and Big Data Engineering (IScIDE 2013) Computational Energy Management in Smart Grids, pp. 161–167. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2014.12.114>. URL: <http://www.sciencedirect.com/science/article/pii/S0925231215009315>.
- [142] F. Wang et al. “Quantized Attention-Gated Kernel Reinforcement Learning for Brain–Machine Interface Decoding”. In: *IEEE Transactions on Neural Networks and Learning Systems* 28.4 (2017), pp. 873–886. DOI: 10.1109/TNNLS.2015.2493079.
- [143] A. Osokin et al. “GANs for Biological Image Synthesis”. In: *CoRR* abs/1708.04692 (2017). arXiv: 1708.04692. URL: <http://arxiv.org/abs/1708.04692>.
- [144] 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.
- [145] H. Gamboa et al. “Electromyography onset detection: new methodology”. In: *Journal of Biomechanics* 45 (2012), S494.
- [146] A. D. Moore. “Synthesized EMG waves and their implications.” In: *American Journal of Physical Medicine & Rehabilitation* 46.3 (1967), p. 1302.
- [147] R. Person and M. Libkind. “Simulation of electromyograms showing interference patterns”. In: *Electroencephalography and clinical neurophysiology* 28.6 (1970), pp. 625–632.
- [148] D. L. Belavý et al. “Analysis of phasic and tonic electromyographic signal characteristics: electromyographic synthesis and comparison of novel morphological and linear-envelope approaches”. In: *Journal of Electromyography and Kinesiology* 19.1 (2009), pp. 10–21.
- [149] I. Murthy and M. Reddy. “ECG synthesis via discrete cosine transform”. In: *Engineering in Medicine and Biology Society, 1989. Images of the Twenty-First Century., Proceedings of the Annual International Conference of the IEEE Engineering in*. IEEE. 1989, pp. 773–774.
- [150] A. Ruha, S. Sallinen, and S. Nissila. “A real-time microprocessor QRS detector system with a 1-ms timing accuracy for the measurement of ambulatory HRV”. In: *IEEE Transactions on Biomedical Engineering* 44.3 (1997), pp. 159–167.
- [151] P. E. McSharry et al. “A dynamical model for generating synthetic electrocardiogram signals”. In: *IEEE Transactions on Biomedical Engineering* 50.3 (2003), pp. 289–294.
- [152] X. Wu and K. Sengupta. “Dynamic Waveform Shaping With Picosecond Time Widths”. In: *IEEE Journal of Solid-State Circuits* (2016).

-
- [153] E. Turajlic. “A novel algorithm for ECG parametrization and synthesis”. In: *Biomedical Engineering and Sciences (IECBES), 2012 IEEE EMBS Conference on*. IEEE. 2012, pp. 927–932.
- [154] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk. “Wavelet-based statistical signal processing using hidden Markov models”. In: *IEEE Transactions on signal processing* 46.4 (1998), pp. 886–902.
- [155] J.-C. Nunes and A. Nait-Ali. “Hilbert transform-based ECG modeling”. In: *Biomedical Engineering* 39.3 (2005), pp. 133–137.
- [156] R. Rodríguez et al. “Hilbert-Huang transform and neural networks for electrocardiogram modeling and prediction”. In: *Natural Computation (ICNC), 2014 10th International Conference on*. IEEE. 2014, pp. 561–567.
- [157] H Atoui, J Fayn, and P Rubel. “A neural network approach for patient-specific 12-lead ECG synthesis in patient monitoring environments”. In: *Computers in Cardiology, 2004*. IEEE. 2004, pp. 161–164.
- [158] M. Merone et al. “ECG databases for biometric systems: A systematic review”. In: *Expert Systems with Applications* 67 (2017), pp. 189–202. ISSN: 09574174. DOI: 10.1016/j.eswa.2016.09.030. URL: <http://dx.doi.org/10.1016/j.eswa.2016.09.030>.
- [159] N. Iyengar et al. “Age-related alterations in the fractal scaling of cardiac interbeat interval dynamics”. In: *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology* 271.4 (1996), R1078–R1084.
- [160] M. Tantawi, K. Revett, and A.-b. Salem. “ECG based Biometric Recognition using Wavelets and RBF Neural Network 1 Introduction 2 Related Work”. In: *7th European Computing Conference* (2013), pp. 100–104.
- [161] M. M. Tantawi et al. “Fiducial feature reduction analysis for electrocardiogram (ECG) based biometric recognition”. In: *Journal of Intelligent Information Systems* 40.1 (2013), pp. 17–39. ISSN: 09259902. DOI: 10.1007/s10844-012-0214-7.
- [162] F. Gargiulo et al. “Subject identification via ECG fiducial-based systems: Influence of the type of QT interval correction”. In: *Computer Methods and Programs in Biomedicine* 121.3 (2015), pp. 127–136. ISSN: 18727565. DOI: 10.1016/j.cmpb.2015.05.012. URL: <http://dx.doi.org/10.1016/j.cmpb.2015.05.012>.
- [163] J. Shen et al. “The PLR-DTW method for ECG based biometric identification”. In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS* (2011), pp. 5248–5251. ISSN: 1557170X. DOI: 10.1109/IEMBS.2011.6091298.
- [164] E. Rabhi and Z. Lachiri. “Biometric personal identification system using the ECG signal”. In: *Procedings in Computing in Cardiology Conference (CinC), 2013*. IEEE. 2013, pp. 507–510.

- [165] K. A. Sidek, V. Mai, and I. Khalil. “Data mining in mobile ECG based biometric identification”. In: *Journal of Network and Computer Applications* 44 (2014), pp. 83–91. ISSN: 10958592. DOI: 10.1016/j.jnca.2014.04.008. URL: <http://dx.doi.org/10.1016/j.jnca.2014.04.008>.
- [166] C. Ye, B. V. Kumar, and M. T. Coimbra. “Heartbeat classification using morphological and dynamic features of ECG signals”. In: *IEEE Transactions on Biomedical Engineering* 59.10 (2012), pp. 2930–2941.
- [167] H. P. Da Silva et al. “Check Your Biosignals Here: A new dataset for off-the-person ECG biometrics”. In: *Computer methods and programs in biomedicine* 113.2 (2014), pp. 503–514.
- [168] A. Lourenço, H. Silva, and A. Fred. “ECG-based biometrics: A real time classification approach”. In: *IEEE International Workshop on Machine Learning for Signal Processing, MLSP* (2012), pp. 1–6. ISSN: 21610363. DOI: 10.1109/MLSP.2012.6349735.
- [169] U. Satija, B. Ramkumar, and M. S. Manikandan. “Automated ECG Noise Detection and Classification System for Unsupervised Healthcare Monitoring”. In: *IEEE Journal of Biomedical and Health Informatics* 22.3 (2018), pp. 722–732. DOI: 10.1109/JBHI.2017.2686436.
- [170] J. N. John, C. Galloway, and A. Valys. “Deep Convolutional Neural Networks for Noise Detection in ECGs”. In: *arXiv preprint arXiv:1810.04122* (2018).
- [171] U. R. Acharya et al. “Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network”. In: *Information Sciences* 405 (2017), pp. 81–90. ISSN: 0020-0255. DOI: 10.1016/J.INS.2017.04.012. URL: <https://www.sciencedirect.com/science/article/pii/S0020025517306539>.
- [172] J. Rodrigues et al. “SSTS: A syntactic tool for pattern search on time series”. In: *Information Processing & Management* 56.1 (2019), pp. 61–76.
- [173] G. Huang, Z. Liu, and K. Q. Weinberger. “Densely Connected Convolutional Networks”. In: *CoRR* abs/1608.06993 (2016). arXiv: 1608.06993. URL: <http://arxiv.org/abs/1608.06993>.
- [174] A. Graves. “Generating Sequences With Recurrent Neural Networks”. In: *CoRR* abs/1308.0850 (2013). URL: <http://arxiv.org/abs/1308.0850>.
- [175] A. Frome et al. “Devise: A deep visual-semantic embedding model”. In: *Advances in neural information processing systems*. 2013, pp. 2121–2129.
- [176] C. Doersch. “Tutorial on variational autoencoders”. In: *arXiv preprint* (2016).
- [177] F. Zhuang et al. “A Comprehensive Survey on Transfer Learning”. In: *arXiv preprint arXiv:1911.02685* (2019).
- [178] G. Lample et al. “Fader networks: Manipulating images by sliding attributes”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 5967–5976.

-
- [179] 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.
- [180] 2019. URL: <https://zenodo.org/>.
- [181] T. Lugovaya. “Biometric human identification based on electrocardiogram”. In: *LETI* (2005).
- [182] G. Ramos et al. “Fatigue Evaluation through Machine Learning and a Global Fatigue Descriptor”. In: *Journal of healthcare engineering* 2020 (2020).
- [183] M. Aronoff and K. Fudeman. *What is morphology*. 2nd ed. Vol. 8. John Wiley & Sons, 2011. ISBN: 978-1-4051-9467-9.
- [184] J. Serra and P. Soille. *Mathematical morphology and its applications to image processing*. 2nd ed. Vol. 2. Springer Science and Business Media, 2012.
- [185] T. T. D. Team et al. “Theano: A Python framework for fast computation of mathematical expressions”. In: *arXiv preprint arXiv:1605.02688* (2016).
- [186] A. K. Jain, P. Flynn, and A. A. Ross. *Handbook of biometrics*. Secaucus, NJ, USA: Springer-Verlag New York, Inc, 2007.
- [187] J. A. Unar, W. C. Seng, and A. Abbasi. “A review of biometric technology along with trends and prospects”. In: *Pattern Recognition* 47.8 (2014), pp. 2673–2688. ISSN: 00313203. DOI: 10.1016/j.patcog.2014.01.016. URL: <http://dx.doi.org/10.1016/j.patcog.2014.01.016>.
- [188] E. L. Van Den Broek. “Beyond biometrics”. In: *Procedia Computer Science* 1.1 (2010), pp. 2511–2519.
- [189] A. Nait-Ali. “Hidden biometrics: Towards using biosignals and biomedical images for security applications”. In: *7th International Workshop on Systems, Signal Processing and their Applications, WoSSPA 2011* (2011), pp. 352–356. DOI: 10.1109/WOSSPA.2011.5931509.
- [190] Y. Sun et al. “Demographic Analysis from Biometric Data: Achievements, Challenges, and New Frontiers”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.2 (2018), pp. 332–351. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2017.2669035.
- [191] R. Snelick et al. “Large-scale evaluation of multimodal biometric authentication using state-of-the-art systems”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.3 (2005), pp. 450–455. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2005.57.
- [192] Z. Wu et al. “ASVspoof: The automatic speaker verification spoofing and countermeasures challenge”. In: *IEEE Journal on Selected Topics in Signal Processing* 11.4 (2017), pp. 588–604. ISSN: 19324553. DOI: 10.1109/JSTSP.2017.2671435.

- [193] R. Palaniappan and D. P. Mandic. “Biometrics from Brain Electrical Activity: A Machine Learning Approach”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.4 (2007), pp. 738–742. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2007.1013.
- [194] A. Nait-Ali. “Hidden biometrics: Towards using biosignals and biomedical images for security applications”. In: *Proc. IEEE Int'l Workshop on Systems, Signal Processing and their Applications (WOSSPA)*. IEEE. 2011, pp. 352–356.
- [195] J. Rodrigues, D. Belo, and H. Gamboa. “Noise detection on ECG based on agglomerative clustering of morphological features”. In: *Computers in biology and medicine* 87 (2017), pp. 322–334.
- [196] S. C. Fang and H. L. Chan. “QRS detection-free electrocardiogram biometrics in the reconstructed phase space”. In: *Pattern Recognition Letters* 34.5 (2013), pp. 595–602. ISSN: 01678655. DOI: 10.1016/j.patrec.2012.11.005. URL: <http://dx.doi.org/10.1016/j.patrec.2012.11.005>.
- [197] G. Wübbeler et al. “Verification of humans using the electrocardiogram”. In: *Pattern Recognition Letters* 28.10 (2007), pp. 1172–1175. ISSN: 01678655. DOI: 10.1016/j.patrec.2007.01.014.
- [198] H. P. Da Silva et al. “Finger ECG signal for user authentication: Usability and performance”. In: *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE. 2013, pp. 1–8.
- [199] R. D. Labati et al. “Deep-ECG: Convolutional Neural Networks for ECG biometric recognition”. In: *Pattern Recognition Letters* (2018). ISSN: 0167-8655. DOI: <https://doi.org/10.1016/j.patrec.2018.03.028>. URL: <http://www.sciencedirect.com/science/article/pii/S0167865518301077>.
- [200] F. Chollet et al. “Keras: Deep learning library for theano and tensorflow”. In: URL: <https://keras.io/k7.8/> (2015), T1.
- [201] *TensorFlow*. URL: <https://www.tensorflow.org/> (visited on 01/31/2019).
- [202] T. K. Roy and M. Morshed. “Performance analysis of low pass FIR filters design using Kaiser, Gaussian and Tukey window function methods”. In: *2013 2nd International Conference on Advances in Electrical Engineering (ICAEE)*. IEEE. 2013, pp. 1–6.
- [203] R. Tan and M. Perkowski. “ECG Biometric Identification Using Wavelet Analysis Coupled with Probabilistic Random Forest”. In: *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*. IEEE. 2016, pp. 182–187.
- [204] A. Page, A. Kulkarni, and T. Mohsenin. “Utilizing deep neural nets for an embedded ECG-based biometric authentication system”. In: *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*. IEEE. 2015, pp. 1–4.

- [205] R. Salloum and C.-C. J. Kuo. “ECG-based biometrics using recurrent neural networks”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE. 2017, pp. 2062–2066.
- [206] Q. Zhang, D. Zhou, and X. Zeng. “Machine learning-empowered biometric methods for biomedicine applications”. In: *AIMS Med Sci* 4 (2017), pp. 274–290.
- [207] S. Bai, J. Z. Kolter, and V. Koltun. “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling”. In: *arXiv preprint arXiv:1803.01271* (2018).
- [208] Y. Wang et al. “Analysis of Human Electrocardiogram for Biometric Recognition”. In: *EURASIP Journal on Advances in Signal Processing* 2008.1 (2008), p. 148658. ISSN: 1687-6180. DOI: 10.1155/2008/148658. URL: <http://asp.eurasipjournals.com/content/2008/1/148658>.
- [209] S. Z. Fatemian and D. Hatzinakos. “A new ECG feature extractor for biometric recognition”. In: *DSP 2009: 16th International Conference on Digital Signal Processing, Proceedings* (2009), pp. 1–6. DOI: 10.1109/ICDSP.2009.5201143.
- [210] R. Silipo and C. Marchesi. “Artificial neural networks for automatic ECG analysis”. In: *IEEE Transactions on Signal Processing* 46.5 (1998), pp. 1417–1425. ISSN: 1053-587X. DOI: 10.1109/78.668803.
- [211] B. H. Brown. *Medical physics and biomedical engineering*. Medical Science Series. Para consulta. Bristol: Institute of Physics Publishing, cop. 1999.
- [212] M. Ishii. *The Neuroscience of Sleep and Dreams*. Yale Journal of Biology and Medicine, 2019.

