



Instituto Superior de Engenharia

Politécnico de Coimbra

DEPARTAMENTO DE / DEPARTMENT OF
INFORMÁTICA E SISTEMAS

IoT Platform for Personal Healthcare of Elderly People

Internship Report para a obtenção do grau de Mestre em / to fulfill
the Master's degree in Informatics Engineering

Branch of Software Engineering

Autor / Author

Ruben António de Jesus Marques

Supervisor

Francisco Pereira

Advisors in the organisation Kirepo S.à.r.l

Steve Reisdorf / Tom Weber

Coimbra, October 2023



INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

ABSTRACT

The global demographic shift towards an ageing population needs innovative approaches to elderly care. As the elderly face complex challenges including declining health and mobility, tailored care solutions are vital. Technological advancements, especially in the realm of the Internet of Things (IoT), offer promising avenues for transforming elderly care. This report encapsulates the creation of an IoT platform designed to address the healthcare needs of elderly individuals. The platform's foundation lies in IoT technology, wearable medical devices, and Domotics. Wearable sensors monitor vital signs, while smart home devices automate tasks, fostering a safer and more independent living environment. The platform's goal is to create individual profiles and detect exceptions in daily routines, initiating appropriate responses when predefined thresholds are reached. Developed during a Master's internship in Software Engineering at the Coimbra Institute of Engineering (ISEC) in Portugal, this initiative collaborates with the Kirepo S.à.r.l company in Luxembourg. It leverages Artificial Intelligence (AI) to analyse sensor data, detect anomalies, and automate workflows, ensuring timely intervention and personalized care. The platform encompasses IoT interactions, APIs, AI-driven triggers, and data analysis. Its adaptability has gained support from healthcare associations in Luxembourg, hinting at the potential for broader implementation beyond the elderly demographic. The internship focuses on a subset of the system, concentrating on key sensors, namely the electrocardiogram (ECG) and motion sensors. ECG sensors detect heart anomalies, while motion sensors track physical activity, detecting sudden changes indicating potential health issues. To assess which is the best way to analyse the sensor data, several Machine Learning techniques are compared and tested such as different feature extraction methods and types of used classifiers (traditional and Neural Networks). For demonstration and validation purposes, a proof of concept is built incorporating the ECG and motion sensors data gathering and the respective analysis of the data. The inclusion of a simplified dashboard permits the visualisation of the data that is gathered and the respective analysis.

Keywords: Elderly Care, Internet of Things, Machine Learning, Electrocardiogram, Human Activity Recognition

RESUMO

A transição demográfica para uma população mais envelhecida a nível global necessita de abordagens inovadoras no cuidado de idosos. À medida que os idosos enfrentam desafios complexos, incluindo saúde e mobilidade decadente, soluções de cuidados adaptadas são vitais. Avanços tecnológicos, especialmente na área da Internet das Coisas (IoT), prometem a transformação do cuidado de idosos. Este relatório resume a criação de uma plataforma IoT projetada para atender às necessidades de saúde dos idosos. A base da plataforma reside na tecnologia IoT, dispositivos médicos vestíveis e domótica. Sensores vestíveis monitorizam sinais vitais, enquanto dispositivos domóticos inteligentes automatizam tarefas, promovendo um ambiente de vida mais seguro e independente. O objetivo da plataforma é criar perfis individuais e detetar exceções nas rotinas diárias, iniciando respostas adequadas quando os limites predefinidos são atingidos. Desenvolvida durante um estágio de Mestrado em Engenharia de Software no Instituto Superior de Engenharia de Coimbra (ISEC) em Portugal, esta iniciativa colabora com a empresa Kirepo S.àr.l no Luxemburgo. A Inteligência Artificial (IA) é usada para analisar dados de sensores, detetar anomalias e automatizar fluxos de trabalho, garantindo intervenção oportuna e atendimento personalizado. A plataforma abrange interações de IoT, interfaces de programação de aplicações (APIs), decisões tomadas por IA e análise de dados. A sua adaptabilidade ganhou o apoio das associações de cuidados no Luxemburgo, sugerindo o potencial para uma implementação mais ampla para além da população idosa. O estágio centra-se num subconjunto do sistema, concentrando-se nos principais sensores, nomeadamente o eletrocardiograma (ECG) e sensores de movimento. Os sensores de ECG detectam anomalias cardíacas, enquanto os sensores de movimento rastreiam a atividade física, detectando mudanças repentinas que indicam possíveis problemas de saúde. Para avaliar qual a melhor forma de analisar os dados dos sensores, diversas técnicas de *Machine Learning* são comparadas e testadas, como diferentes métodos de extração de *features* e tipos de modelos utilizados. Para efeitos de demonstração e validação, é construída uma prova de conceito incorporando a recolha de dados de ECG e sensores de movimento e a respetiva análise dos dados. A inclusão de um *dashboard* simplificado permite a visualização dos dados recolhidos e a respetiva análise.

Palavras-chave: Cuidado de idosos, Internet das Coisas, Machine Learning, Eletrocardiograma, Reconhecimento de Atividade Humana

EPIGRAPH

In computer science, as in life, the “80/20 Rule” unveils the essence: 20% of the efforts yield 80% of the results.

ACKNOWLEDGEMENT

First and foremost, I express my sincere gratitude to the Coimbra Institute of Engineering, along with its academic staff and professors, for their pivotal role in facilitating the completion of my Bachelor's degree and the pursuit of my Master's degree, which is the subject of this document. I reserve a special dedication to my PhD supervisor, Professor Francisco Pereira, whose unwavering support and guidance were vital throughout the course of this internship project. I am profoundly thankful for his invaluable insights, particularly in the field of Machine Learning, and for his indispensable assistance during the composition of this Master's thesis, to which a significant portion of the credit for its success is due.

Secondly, I wish to express my appreciation to Kirepo for allowing me to engage in an internship centred on a multifaceted project related to IoT and elderly care. I am equally thankful for the autonomy granted to me in the project's development and research. In particular, I would like to acknowledge Steve Reisdorf, CEO, and Tom Weber for their consistent support and flexibility over the duration of the internship.

A profound debt of gratitude is owed to my girlfriend, Gabriela Veríssimo, who has been my constant companion throughout my academic journey, culminating in the completion of this Master's thesis. Her own successful Master's thesis defence served as a significant source of inspiration and motivation for the development of my thesis. I am deeply thankful for her illimited support across all facets of my life, especially during the conclusion of my Master's degree.

I wish to extend my appreciation to André Santos, with whom I shared a productive and mutually supportive working relationship throughout the entire process of this Master's course. Our collaboration, marked by a spirit of healthy competition, played a pivotal role in our collective success.

To conclude, I wish to offer my heartfelt thanks to my family, particularly my parents, for their enduring belief in my abilities and unwavering faith in my potential.

TABLE OF CONTENTS

Abstract.....	i
Resumo	ii
Epigraph	iii
Acknowledgement.....	iv
Table of Contents.....	v
List of Figures	viii
List of Tables.....	x
List of Abbreviations	xi
1 Introduction.....	1
1.1 Coimbra Institute of Engineering	1
1.2 Kirepo S.à.r.l	2
1.3 The emergence of wearable and domotic technology	2
1.4 Scope of the internship.....	3
1.5 Objectives and contributions.....	4
1.6 Structure of the document	4
2 Technological background.....	6
2.1 Internet of Things	6
2.1.1 Healthcare and medical devices.....	6
2.1.2 Domotics.....	6
2.2 Communication	7
2.2.1 Short-range technologies.....	7
2.2.2 Long-range technologies	7
2.2.3 Message brokers	7
2.3 Artificial Intelligence	8
2.3.1 Machine Learning.....	9
2.3.2 Metrics	15
3 General Platform architecture.....	18
4 Related work.....	21
4.1 Domotic and home-oriented solutions	21
4.2 Healthcare-oriented solutions.....	22
5 ECG analysis	24

5.1	Fundamentals of Electrocardiography.....	24
5.1.1	ECG curve	24
5.1.2	Lead system.....	25
5.1.3	ECG devices	26
5.2	Machine Learning for ECG analysis	26
5.2.1	Related work	26
5.2.2	Dataset discovery.....	27
5.2.3	Pre-processing.....	30
5.2.4	Feature extraction.....	32
5.2.5	Model training.....	34
5.2.6	Results and discussion	37
6	Motion analysis.....	43
6.1	Motion sensors	43
6.1.1	Accelerometer.....	43
6.1.2	Gyroscope	44
6.1.3	GPS	44
6.2	Machine Learning for motion analysis.....	44
6.2.1	Related work	45
6.2.2	Dataset creation.....	45
6.2.3	Pre-processing and feature extraction.....	47
6.2.4	Model training.....	49
6.2.5	Results and discussion	49
7	Implementation of the Elderly Care Platform	52
7.1	Scope of the proof of concept.....	52
7.1.1	Features.....	52
7.1.2	Requirements	53
7.2	Materials and Methods.....	56
7.2.1	Python.....	56
7.2.2	Vue.js.....	56
7.2.3	Android Studio	57
7.2.4	MQTT.....	57
7.2.5	Software design patterns.....	57
7.3	General System Architecture	58

IoT Platform for Personal Healthcare of Elderly People

7.3.1	Central MQTT Broker.....	60
7.3.2	The Orchestrator module.....	60
7.3.3	IoT sensors and devices	61
7.4	ECG recording and analysis workflow	62
7.5	Motion recording and analysis workflow	63
7.6	ECG control workflow.....	65
7.7	Dashboard.....	66
8	Conclusions and future work	71
8.1	Future work.....	72
	References	74
	Appendix	82
	Appendix A – Internship proposal	83
	Appendix B – ML models’ performance charts of all ML test instances for ECG classification (Feature extraction methods comparison)	86
	Appendix C – ML models’ performance charts of all ML test instances for ECG classification (Model comparison).....	92
	Appendix D – Architecture of the Minas Gerais Neural Network.....	96
	Appendix E – Architecture of the Exploratory Neural Network	97

LIST OF FIGURES

Figure 1 – Depiction of the generic functioning of the publish and subscribe pattern. Image retrieved from [8].....	8
Figure 2 – Illustration of the relationship between Artificial Intelligence, Statistics, Machine Learning, Deep Learning and Expert Systems. Image retrieved from [10].	9
Figure 3 – Breakdown illustration of Machine Learning in several branches. Image retrieved from [13].....	10
Figure 4 – Depiction of the model creation process. Image retrieved from [14]....	11
Figure 5 – Simple illustration of how a neuron works in Deep Learning and comparing it to a biological neuron (right bottom corner). Image adapted from [20].	12
Figure 6 – Depiction of some of the types of possible ANN architectures. Image retrieved from [21].....	13
Figure 7 – How DL is a subset of ML and how ML is a subset of AI. Image retrieved from a Wikipedia article about Deep Learning [22].....	14
Figure 8 – Example of a confusion matrix about ECG predictions. Image generated with [27].	15
Figure 9 - Relationship between the precision and recall metrics. Image retrieved from Wikipedia [30].	17
Figure 10 – General platform architecture diagram. Produced with draw.io software [32] and Flaticons [33].	19
Figure 11 – Random normal 10-second ECG recording from the DeepFake database [46]. Chart generated with Python’s “ecg-plot” library.....	24
Figure 12 – Characterization of an ECG heartbeat segment with the five labelled waves from left to right (S, QRS complex and T). Image retrieved from a Wikipedia article about electrocardiography [47].....	25
Figure 13 – Representation of the three limb leads (I, II and III) and their corresponding ECG waveforms following the Einthoven triangle. Image retrieved from a Wikipedia article about electrocardiography [47].....	25
Figure 14 – Diagram explaining how the uniformisation for all datasets is performed. Produced with draw.io software [32] and Flaticons [33].	31
Figure 15 – Performance metrics of the Random Forest classifier.....	50
Figure 16 – Confusion matrix of the predicted labels of inactive, active and walking activities.	51

Figure 17 – General architecture diagram. Produced with draw.io software [32] and Flaticons [33]..... 59

Figure 18 – ECG recording and analysis workflow diagram. Produced with draw.io software [32] and Flaticons [33]..... 63

Figure 19 – Motion recording and analysis workflow diagram. Produced with draw.io software [32] and Flaticons [33]..... 65

Figure 20 – ECG control workflow diagram. Produced with draw.io software [32] and Flaticons [33]. 66

Figure 21 – Dashboard and MQTT communication diagram. Produced with draw.io software [32] and Flaticons [33]..... 68

Figure 22 – Screenshot of the simple GUI to visualize ECG and motion data..... 69

LIST OF TABLES

Table 1 – List of datasets used in this project for ECG analysis.....	29
Table 2 – List of the five feature vector creation methods.....	34
Table 3 – List of classifiers submitted to test in this study	36
Table 4 – Summary of key comparative results (unbalanced dataset only).....	39
Table 5 – Description of each recorded activity.....	46
Table 6 – Description of the variables that were gathered from each sensor	47
Table 7 – List of features used to feed the ML classifiers.....	48
Table 8 – List of requirements.....	53

LIST OF ABBREVIATIONS

AI – Artificial Intelligence
ANN – Artificial Neural Network
BLE – Bluetooth Low-Energy
CSV – Comma-Separated Values
DL – Deep Learning
DNN – Deep Neural Network
EBM – Explainable Boosting Machine
ECG – Electrocardiogram
GPS – Global Positioning System
GUI – Graphical User Interface
HAR – Human Activity Recognition
HDF5 – Hierarchical Data Format 5
IoT – Internet of Things
KNN – k-Nearest Neighbours
ML – Machine Learning
MQTT – Message Queuing Telemetry Transport
NFC – Near Field Communication
PoC – Proof of concept
RAM – Random access memory
SVM – Support Vector-Machine
WFDB – WaveForm Database

1 INTRODUCTION

The world is experiencing a significant demographic shift with an ageing population that is becoming a prevailing global trend. In 2019, the size of this demographic surpassed 1 billion, and projections indicate a rise to 2.1 billion by 2050 [1]. Some projections suggest a tripling in the age group of 80 years and above, from 143 million in 2019 to 426 million by 2050 [2]. As life expectancy rises, the challenges of elderly care are becoming increasingly complex and demanding for societies worldwide. Providing comprehensive and efficient care for elderly individuals has become a pressing concern for governments, healthcare institutions, families, and caregivers alike.

The elderly face unique challenges, including declining mobility, cognitive abilities, and potential health complications, necessitating tailored care and constant monitoring. Family members and healthcare facilities often cope with the pressure of the escalating demand for elderly care. However, the shortage of trained professionals, the emotional impact on seniors, and the increasing demand for specialized facilities pose significant obstacles.

Recognizing the importance of supporting elderly individuals in maintaining their independence and quality of life, technological advancements have emerged as a promising solution. The convergence of the Internet of Things (IoT), wearable medical devices, and Domotics opens new possibilities for empowering elderly individuals to age in place, enabling them to reside in their homes longer without the need for constant external assistance.

To this extent, the present report details the conception of an IoT platform for the personal healthcare of elderly people that has been generated during a second-year Master's internship in Software Engineering at the Coimbra Institute of Engineering (ISEC) in Portugal. This initiative is conducted in collaboration with the Kirepo S.à.r.l company based in Luxembourg.

1.1 Coimbra Institute of Engineering

The Coimbra Institute of Engineering (ISEC) [3], situated on the right bank of the Mondego River, forms a cornerstone of the Coimbra Polytechnic Institute (IPC). With a legacy spanning over four decades in student education and development, ISEC originated in 1974 through the transformation of the former Coimbra Industrial and Commercial Institute, established in December 1921. ISEC's commitment to delivering excellence in engineering education is coupled with its aspiration to stand as a beacon of distinction on both the national and international educational landscapes. Rooted in its mission of "the creation, transmission, and

dissemination of culture, science, and technology", ISEC provides advanced education tailored to professional engagements within the field of engineering. Notably, ISEC maintains robust ties with diverse organizations, fostering an environment of enhanced interaction between students and companies. This synergy not only enriches the academic experience but also bolsters success within the competitive job market. ISEC provides a comprehensive range of engineering courses, including both Bachelor's and Master's degrees in fields like civil engineering, electrical engineering, computer science and mechanical engineering. This report happens in the context of the Master's in Computer Science program, which is designed to elevate computer science education. Its primary goal is to produce professionals equipped with engineering expertise for the design, development, and maintenance of computer systems across various industries. This program offers two specializations: Software Engineering and Intelligent Data Analysis; and this report happens in the context of the branch of Software Engineering.

1.2 Kirepo S.à.r.l

Established in 2004, Kirepo is a Luxembourgish company that has consistently pioneered tailored innovations fostering business growth [4]. Leveraging cutting-edge technologies, such as versatile applications accessible across various platforms, Kirepo empowers businesses by enhancing engagement and operational efficiency. At the core of Kirepo's suite of offerings lie adaptable applications, facilitating effective product marketing and seamless information dissemination. These solutions are enhanced by advanced features, accommodating modern digital landscapes across devices. Moreover, Kirepo addresses diverse requirements with its array of tools, providing a versatile platform for dynamic content management. Complementing these initiatives, Kirepo's dedication to holistic solutions extends to facilitating streamlined transactions and supply requisition processes. The company's journey is marked by collaborative partnerships with prominent clients and close relations with governmental projects, embodying innovation as the driving force propelling advancement.

1.3 The emergence of wearable and domotic technology

The realm of technology has witnessed a significant transformation with the emergence of wearable IoT technologies and Domotics. These advancements offer invaluable tools for monitoring vital signs and activities through wearable sensors and enable the automation of routine tasks through smart home devices.

Wearable technologies encompass a wide range of medical sensors and IoT devices. Key sensors include the electrocardiogram (ECG) sensor, which provides insights into heart health, heart rate monitoring for stress assessment, and sensors for

respiration, heat, motion, sleep, and sound, facilitating anomaly detection and sleep disorder evaluation.

In parallel, Domotics integrates sensors and actuators to automate tasks and enhance home functionality. Automated lighting, window control, and security cameras ensure a comfortable and secure living environment.

The synergy between wearable technologies and Domotics opens new possibilities for comprehensive monitoring and prompt responses to potential issues. These advancements pave the way for improved independence and safety, offering personalized and efficient solutions to meet the evolving needs of individuals.

In conclusion, the emergence of wearable technologies and Domotics revolutionizes the landscape of technology, offering seamless integration for enhanced monitoring and automation, providing a transformative impact on various aspects of modern living.

1.4 Scope of the internship

Kirepo is at the forefront of developing an innovative IoT-based platform designed to support elderly people in their homes. By leveraging interconnected devices, the goal of the platform is to create individual profiles and identify exceptions in their daily routines. When a predefined threshold is reached, the platform initiates appropriate responses, such as contacting a call centre, alerting family members, or summoning professional assistance.

The challenge lies in seamlessly communicating with diverse IoT sensors, effectively assessing the status of elderly individuals, and interacting with them in real-time. The platform orchestrates the interaction between various sensors and can pilot other devices, such as enabling remote door opening, automatizing lights, and facilitating person-to-person dialogues.

A key aspect of the platform involves the use of Artificial Intelligence (AI) to analyse sensor data, detect anomalies, and automate workflows. This AI-driven analysis enables the identification of potential health issues promptly, ensuring timely intervention and personalized care.

The scope of the platform encompasses various computer science disciplines, including IoT interactions, APIs, AI-driven triggers, and data analysis. Healthcare associations in Luxembourg have warmly received the concept, expressing their support for its potential impact on the elderly care landscape. The platform's adaptability and positive reception open the possibility for future expansion to other countries and populations beyond the elderly demographic.

The focus of this internship is to develop a proof of concept (PoC) for a subset of the aforementioned system, concentrating on key sensors, namely the ECG and motion sensors such as the accelerometer, gyroscope and Global Positioning System

(GPS). The ECG sensor plays a pivotal role in monitoring the electrical activity of the heart, allowing for the early detection of anomalies such as arrhythmias or heart attacks. On the other hand, motion sensors track the physical activity of the individual, providing insights into movement patterns and enabling the detection of sudden changes that may indicate potential health issues.

Ultimately, the development of this IoT-based elderly care platform holds immense promise in revolutionizing elderly care practices, fostering independence, and enhancing the overall quality of life for senior citizens. As technological advancements continue to accelerate, the platform's expansion and incorporation of additional sensors and capabilities can further bolster its effectiveness and contributions to the evolving landscape of elderly care.

The original and detailed internship proposal can be found in Appendix A for further information.

1.5 Objectives and contributions

This internship aims to assess the feasibility of utilizing IoT sensors, including medical wearables and home automation devices, to create a healthcare platform for continuous monitoring of the elderly, thereby improving their daily lives. Within the broad field of IoT technologies, this project focuses on evaluating specific sensors and devices to develop a PoC and assess its potential for future development.

To this extent, the goals that are pursued in this project consist in:

- Conducting a state-of-the-art analysis of IoT devices in the healthcare sector.
- Performing a state-of-the-art analysis of analogous projects related to elderly care.
- Defining the scope of the PoC.
- Applying Machine Learning (ML) techniques to analyze ECG and motion data.
- Implementing the software components of the PoC.
- Validating the PoC.

1.6 Structure of the document

This document is structured into eight chapters, each serving a distinct purpose within the context of the internship project. The first chapter introduces the internship project's scope and primary challenges. The second chapter provides the technological background. The third chapter outlines the general system architecture and high-level requirements. The fourth chapter examines analogous projects in

elderly care and healthcare with a state-of-the-art analysis. In the fifth chapter, we delve into the ML analysis applied to ECG data. Similarly, the sixth chapter discusses the ML analysis applied to motion data. Building on insights from previous chapters, the seventh chapter focuses on implementing software modules for the PoC. Lastly, the eighth chapter ends with the project conclusions and future perspectives.

2 TECHNOLOGICAL BACKGROUND

The technological landscape of healthcare, particularly in the realm of elderly care, has witnessed a remarkable transformation owing to the fusion of IoT with medical sensor technologies. This convergence has enabled the continuous monitoring of vital health parameters and environmental factors, generating vast streams of data. In tandem with this, ML techniques have been enhanced and tailored to this field, acting as a pivotal tool to analyse and derive meaningful insights from this type of data. To offer contextual insights into the technologies and concepts employed in this project, the following subsections intend to elucidate each of them and elucidate their relevance to the project.

2.1 Internet of Things

The Internet of Things encompasses a range of definitions, but at its core, it signifies a network where devices interact autonomously, enabling seamless data collection and exchange [5]. This innovation drives automation across industries and facilitates the gathering of data about any kind of subject.

IoT encompasses a wide range of sensors and devices, fostering the creation of diverse applications like smart parking systems, precision agriculture, and water usage optimization, among others. Similarly, the array of IoT sensors opens avenues for smart healthcare systems, capitalizing on significant advancements in portable and wearable medical technologies. Taking this into consideration, the prospect of bringing healthcare to patients' homes through a remote care approach becomes increasingly viable since vital signs (e.g., ECG, respiration rate, etc.) that could only be monitored in clinical setups can now be monitored at home with IoT devices.

2.1.1 Healthcare and medical devices

The spectrum of IoT devices and sensors that may be used in healthcare systems varies from sensors that measure the most vital data such as ECG, oxygen saturation, and breathing rhythm to less critical information like blood pressure, and body temperature, among others. The availability of devices that incorporate the mentioned sensors keeps evolving which potentiates more and more research and experimentation on the subject.

2.1.2 Domotics

Parallel to the emergence of healthcare IoT technologies, arises the concept of Domotics which is also referred to as home automation or smart home technology [6]. Several devices are grouped within the Domotics field such as lights, power outlets, blinds, thermostats, and smoke detectors, among others. Those devices

permit gathering information about the environment of a patient other than medical data and also provide automation capabilities to assist the elderly.

2.2 Communication

In the realm of IoT, communication technologies play a pivotal role, providing the fundamental functionality and purpose of the IoT concept. These technologies encompass both short-range and long-range varieties, facilitating the exchange of information among IoT devices and enabling seamless control and interaction with other components within a system.

2.2.1 Short-range technologies

In the realm of short-range communication technologies, Bluetooth stands out as a wireless solution for data exchange over short distances, commonly found in headphones, speakers, and smart home devices. On the other hand, Zigbee and Bluetooth Low-Energy (BLE) offer low-power wireless communication for IoT devices like thermostats and medical sensors. Another short-range option is Near Field Communication (NFC), enabling secure data exchange between nearby devices and finding applications in contactless payments and device pairing.

2.2.2 Long-range technologies

In the context of long-range communication technologies, Wi-Fi emerges as a prominent choice, providing high-speed internet connectivity over larger areas, catering to devices such as smart televisions and cameras. Cellular networks, including 4G and the upcoming 5G, offer extensive coverage and rapid data transmission, making them suitable for applications like asset tracking and remote monitoring. The GPS, a satellite-based navigation system, ensures accurate location information and motion through space, finding utility in the tracking of many objects.

2.2.3 Message brokers

A message broker is a software tool that facilitates communication and information exchange among applications, systems, and services [7]. It acts as an intermediary responsible for delivering the messages sent between the given services. The main advantage of a message broker is that it is agnostic and platform-independent, which permits the intercommunication between services from different programming languages or different platform implementations. This way, services and software components can uncouple the communication logic from their core business logic. The inherent high heterogeneity within IoT's encompassed technologies and devices presents a challenge in creating comprehensive solutions involving diverse components since every device has its way of communicating with the rest of the

system. To this extent, message brokers serve as intermediaries for facilitating message transmission within highly heterogeneous IoT systems, ensuring agnostic communication between diverse devices and technologies.

A couple of message distribution types can be used with message brokers, with the publish and subscribe pattern being the most widely used. This messaging pattern involves message distribution where message producers publish messages on specific topics, while multiple message consumers subscribe to topics of interest. Messages published within a topic are then distributed to all subscribed consumers [7]. Figure 1 illustrates how the generic publish and subscribe pattern works within a system.

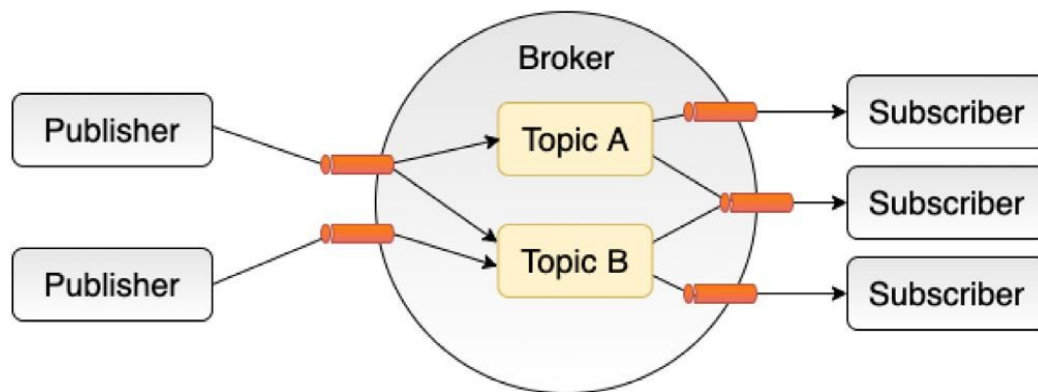


Figure 1 – Depiction of the generic functioning of the publish and subscribe pattern. Image retrieved from [8].

2.3 Artificial Intelligence

Artificial Intelligence was first introduced by John McCarthy in 1955 as "the science and engineering of making intelligent machines" [9]. In essence, AI entails employing methods to enable computer systems to execute human tasks [10]. In Figure 2, the composition of AI can be contemplated along with how it coexists with the branch of statistics. Artificial Intelligence can be further divided into two primary branches: Machine Learning, employing mathematical models to learn from data and discern underlying patterns, and Expert Systems, which consist of autonomously operating rule-based systems created by humans, such as fuzzy logic. However, for this work, the focus will reside on Machine Learning for data analysis.

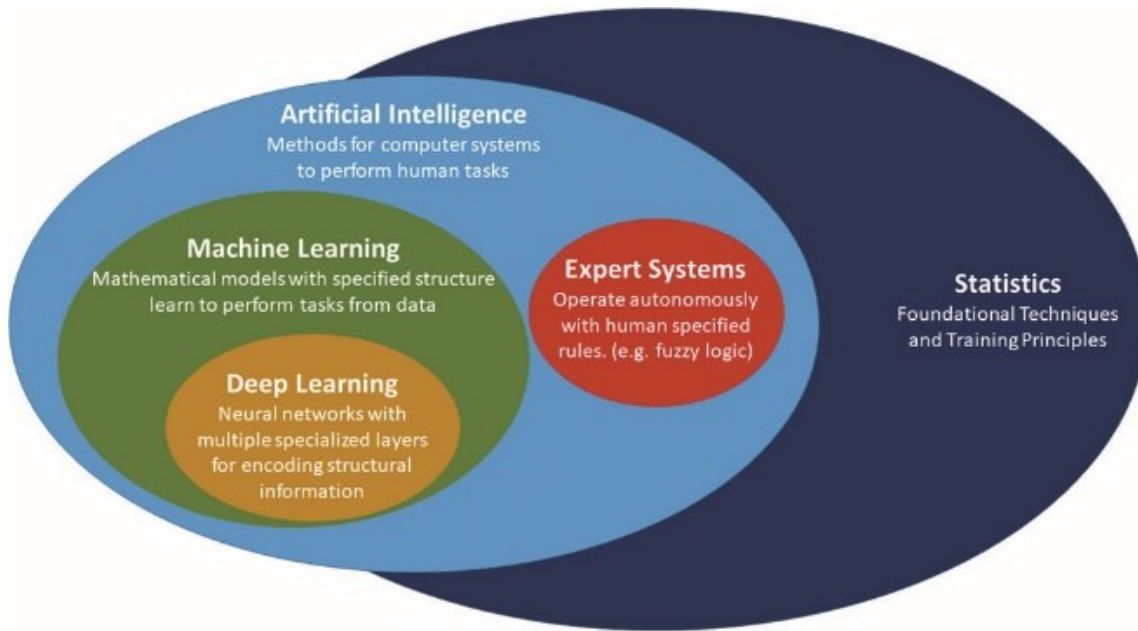


Figure 2 – Illustration of the relationship between Artificial Intelligence, Statistics, Machine Learning, Deep Learning and Expert Systems. Image retrieved from [10].

2.3.1 Machine Learning

The first mention of Machine Learning in the scientific literature was made decades ago by a researcher trying to figure out how to program a computer to play checkers and defined the field of ML as “a field of study that gives computers the ability to learn without being explicitly programmed” [11].

ML algorithms are computational procedures that function through the utilization of input data to accomplish designated tasks. In contrast to conventional programming paradigms characterized by deterministic directives (e.g. conditional or fuzzy logic), these algorithms adopt a more flexible approach. Their capability of learning through repetition and previous experiences permits them to be used in a variety of contexts in society [12].

Figure 3 illustrates the three primary domains of Machine Learning: Supervised Learning, Unsupervised Learning, and Reinforcement Learning, each addressing unique challenges and characteristics. Supervised Learning algorithms work with datasets comprising features, with each input example linked to a corresponding label or target output. Unsupervised Learning algorithms analyze datasets with numerous features to extract structural properties, aiming to uncover patterns without prior knowledge of underlying labels, unlike Supervised Learning. Reinforcement Learning employs a self-teaching framework based on dynamic and systematic learning through actions and subsequent rewards, commonly used in games and robotics [13].

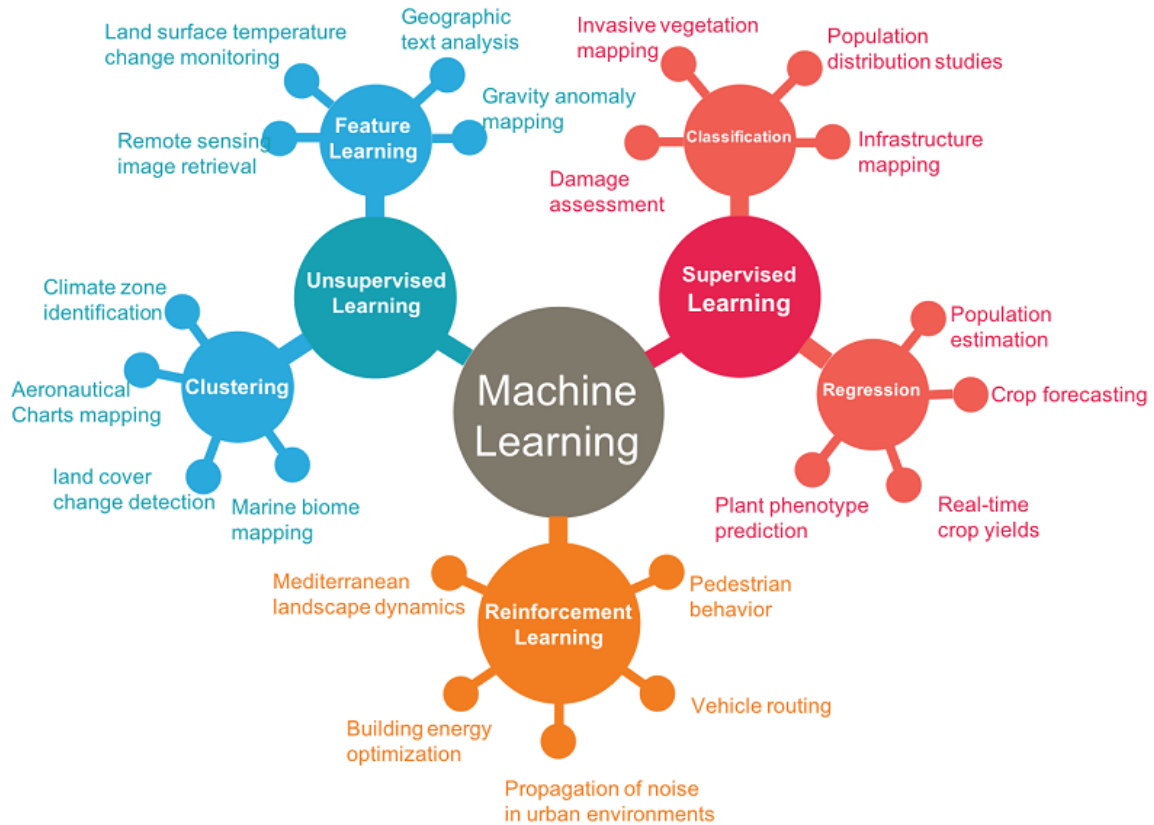


Figure 3 – Breakdown illustration of Machine Learning in several branches. Image retrieved from [13].

2.3.1.1 Supervised Learning

In the realm of Machine Learning, various learning methodologies are available to facilitate the creation of models capable of acquiring knowledge from data. Supervised Learning represents one of the learning approaches and finds application in approximating an unknown mapping of (input, output) pairs based on known (input, output) samples [12]. To elaborate, the known mappings embody the training data, comprising labelled inputs, while the unknown mappings encompass data never encountered by the model. Consequently, the model tries to forecast the output for an unfamiliar input by drawing on the training data it has been furnished with. This task is also referred to as classification. In summary, the creation of an ML model comprises three primary stages: data collection to form the training and test datasets, model training using an ML algorithm, and the subsequent validation using data not encountered during the training phase. In the end, the model should be capable of performing future predictions autonomously as a way of being applied in a real-world scenario. Figure 4 depicts the different steps of the model creation process in ML.

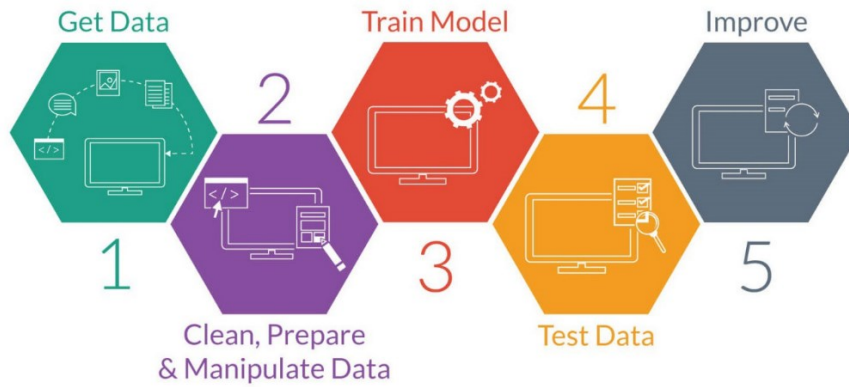


Figure 4 – Depiction of the model creation process. Image retrieved from [14].

In the field of Supervised Learning, various approaches are employed to create models that can train on data, learn from it, and subsequently classify new, unfamiliar instances. There are numerous algorithms available that permit the creation of models for classification, each with unique characteristics and capabilities for addressing different problems. These algorithms include Support Vector Machines, k-Nearest Neighbours, Decision Trees, Random Forests, Logistic Regression, Naïve Bayes, Neural Networks, and others [15]. In this project, our focus resides on Random Forests, k-Nearest Neighbours, Explainable Boosting Machines, and Neural Networks.

Random Forest (RF): Random Forests consist of an ensemble of tree predictors, where each tree relies on the values of a random vector [16]. This vector is independently sampled and follows the same distribution for all trees within the forest. As the number of trees in the forest increases, the generalization error of the forests converges to a limit. The generalization error of a forest of tree classifiers relies on two factors: the individual strength of the trees within the forest and the correlation between them [16].

k-Nearest Neighbour (KNN): The Nearest Neighbour is a simple and powerful technique in Machine Learning that determines the class of a query by identifying its nearest neighbours [17]. This approach is important due to improved computational power, reducing concerns about runtime performance. The KNN variant considers multiple neighbours for classification. Since the training examples are required to be in memory during runtime, this approach is often referred to as Memory-based Classification [17].

Explainable Boosting Machine (EBM): The Explainable Boosting Machine is a type of Generalized Additive Model that uses tree structures and employs cyclic gradient boosting and automatic interaction detection [18]. EBMs offer a balance between accuracy similar to advanced black box models and full interpretability. Although they may require more time for training compared to traditional algorithms, they stand out for their efficiency in compactness and rapid prediction capabilities [18].

2.3.1.1.1 Deep Learning

Deep Learning (DL) is a field of Machine Learning where computational algorithms and models are created that emulate the structural composition of biological neural networks within the brain, known as Artificial Neural Networks (ANN) [19]. The term “deep” indicates that several layers exist within a network. In short, three types of layers exist with them being the input layer, responsible for receiving input data; the output layer, tasked with generating the outcomes of data processing; and the hidden layer, entrusted with the extraction of inherent patterns embedded within the data [19]. When multiple hidden layers exist within an ANN, it is also commonly referred to as a Deep Neural Network (DNN).

Neuron: The neuron stands as a fundamental element within the realm of DL, serving as the primary constituent of ANNs. It receives a set of independent variables as input and generates an output through computations applied to these input variables. Each input variable possesses an associated weight, signifying its contribution to the final output. The neuron then conducts a sum operation on the input values, and the resulting sum is subsequently processed through an activation function that determines the output value [20]. Figure 5 depicts the functioning of an artificial neuron and how it relates to a biological neuron.

The Neuron

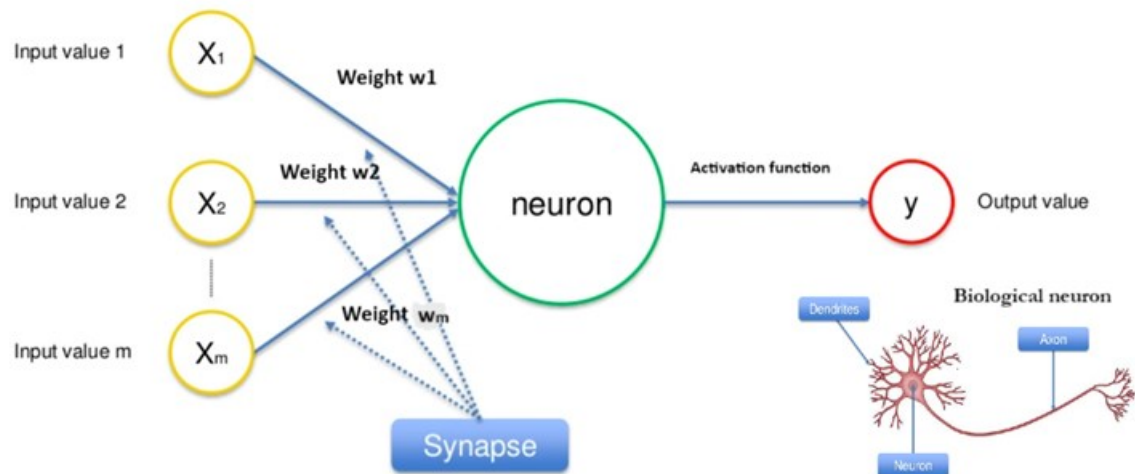


Figure 5 – Simple illustration of how a neuron works in Deep Learning and comparing it to a biological neuron (right bottom corner). Image adapted from [20].

Artificial Neural Network: ANNs consist of multiple layers of interconnected neurons. For the sake of simplicity, they are also commonly referred to as Neural Networks. When there is only one neuron, it is referred to as a perceptron. Various ANNs exhibit distinct architectures, varying in the number of layers and their interconnections. The two primary categories of ANNs are Feed-Forward Networks and Recurrent/Feedback Networks [21]. In Feed-Forward Networks, input information flows in one direction through the layers until it reaches the output layer.

In contrast, Recurrent/Feedback Networks involve bidirectional communication, where input is exchanged back and forth in multiple iterations until it reaches the output layer.

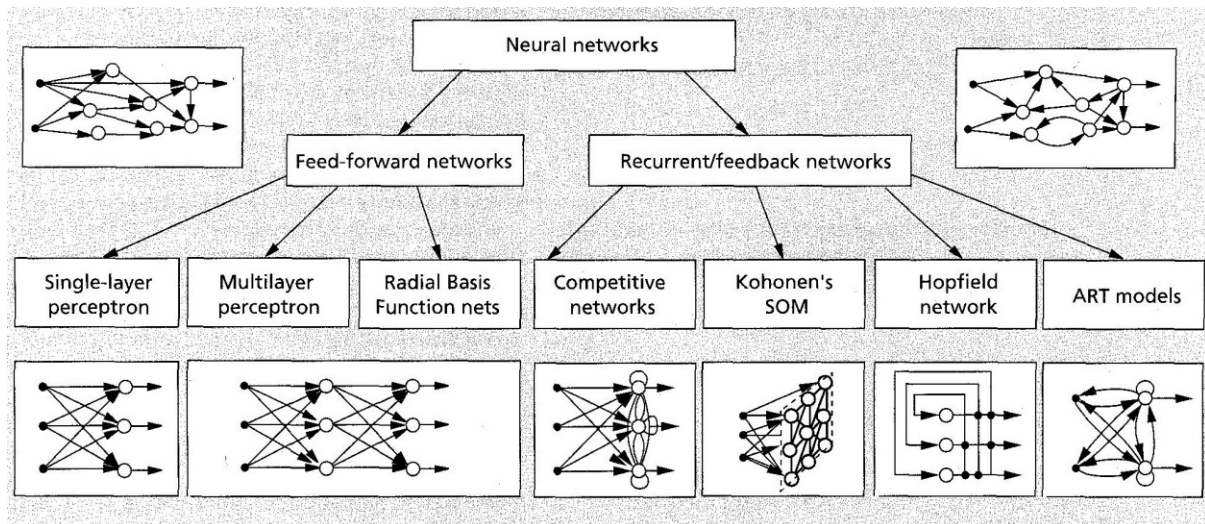


Figure 6 – Depiction of some of the types of possible ANN architectures. Image retrieved from [21].

DL permits to steer Supervised Learning techniques, such as classification tasks, on data such as images, and sounds, among others as a way of resolving problems that involve image or speech recognition for example. On the contrary, more conventional ML models encounter limitations when dealing with raw, unprocessed natural data, while Deep Learning has the potential to effectively manage the intricacies of this type of data.

Figure 7 helps to depict how Deep Learning integrates the field of Machine Learning in the field of Artificial Intelligence.

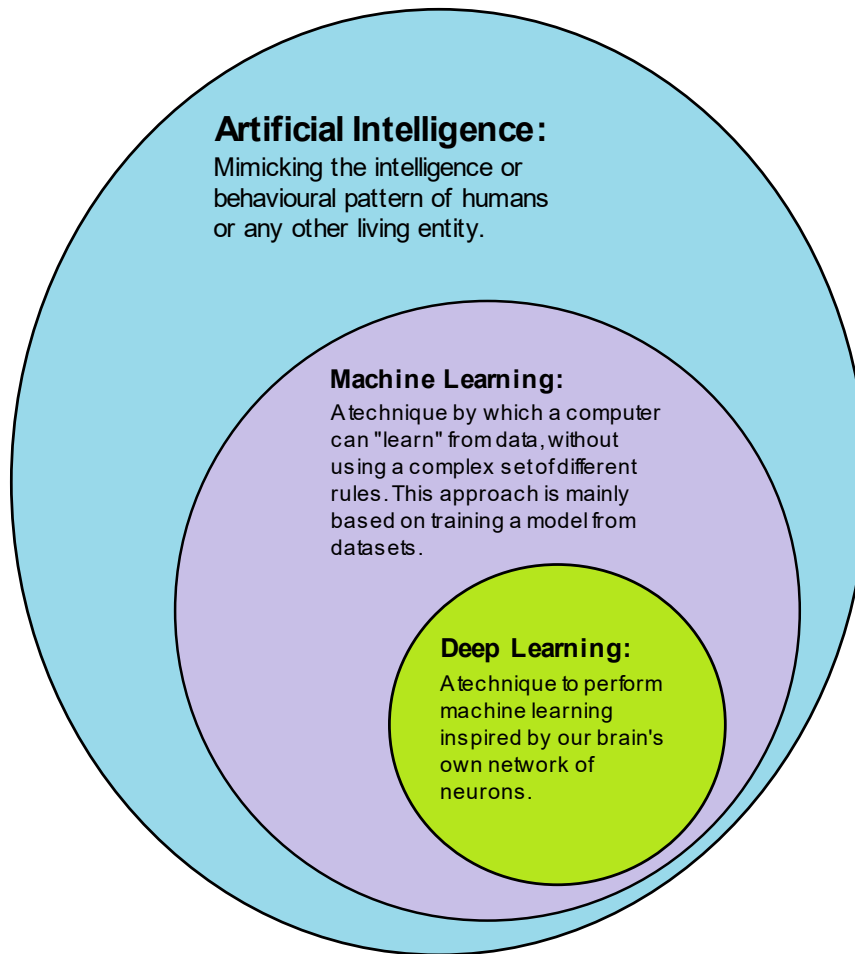


Figure 7 – How DL is a subset of ML and how ML is a subset of AI. Image retrieved from a Wikipedia article about Deep Learning [22].

Convolutional Neural Network (CNN): CNNs emerged from the study of the brain's visual cortex and have been applied to image recognition since the 1820s [23]. It excels at identifying patterns in images, allowing it to recognize objects and categories effectively. CNNs can also be applied to audio, time series, and signal data. These networks can comprise numerous layers, each specializing in detecting different image features. Filters are used to scan images at various resolutions, and the results from these scans feed into subsequent layers. These filters start by identifying basic attributes like brightness and edges and progress to more complex features that uniquely characterize the object.

Residual Neural Network:

A Residual Neural Network, also known as ResNet, is a DL approach where the network's layers are designed to learn what's left over or "residual" from the input data [24]. Those layers are called residual units. They enable the training of very deep DNNs by addressing the vanishing gradient issue, leading to improved feature learning and superior performance on tasks like image classification and object detection. ResNets are also excellent for transfer learning, allowing pre-trained models to be fine-tuned for specific tasks. Their interpretability makes them a

preferred choice for many applications, making them a substantial advancement over traditional CNNs [23].

2.3.2 Metrics

When working with ML models to perform classification tasks, it is important to have assessment metrics that measure the performance of a given classifier. Various metrics, including accuracy, precision, and recall, among others, are employed in Machine Learning to gauge model performance. The significance of these metrics varies based on the specific objectives they aim to fulfil. In this project, the subsequent metrics are important as a way of evaluating how well the models perform their task of evaluating the normality of ECG samples or motion activities later in this work.

2.3.2.1 Confusion matrix

In ML, a confusion matrix consists of a table that serves as a visual tool for assessing the effectiveness of an algorithm, typically in the context of Supervised Learning [25]. In this matrix, every row corresponds to instances within an observed class, and each column pertains to instances classified into a predicted class [26]. This permits to assess the performance of the classifiers within each class. Figure 8 depicts an example of a confusion matrix about an arbitrary and random binary ECG classification. In this instance, the model correctly predicted 120 normal samples and erroneously identified 5 normal samples as abnormal. Conversely, it also incorrectly classified 8 abnormal samples as normal, while accurately predicting 80 genuine abnormal samples. These observations provide the basis for calculating additional performance metrics, including accuracy, precision, recall, and the F1-score.

ECG prediction			
TARGET \ OUTPUT	NORMAL	ABNORMAL	SUM
NORMAL	120 56.34%	8 3.76%	128 93.75% 6.25%
ABNORMAL	5 2.35%	80 37.56%	85 94.12% 5.88%
SUM	125 96.00% 4.00%	88 90.91% 9.09%	200 / 213 93.90% 6.10%

Figure 8 – Example of a confusion matrix about ECG predictions. Image generated with [27].

2.3.2.2 Accuracy

The accuracy metric is used to measure the proportion of correctly predicted instances (both true positives and true negatives) out of the total instances in a dataset, indicating how well a model predicts across all classes. The following formula defines how the accuracy is calculated [28]:

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

where TP = True positive; FP = False positive; TN = True negative; FN = False negative

2.3.2.3 Precision

The precision metric measures the accuracy of positive predictions made by the model. It is calculated as the ratio of true positive predictions (correctly identified positive cases) to the sum of true positive and false positive predictions (incorrectly identified positive cases). For example, in ECG classification, precision indicates how well the model correctly identifies actual cases of the target condition (e.g., abnormal heart rhythms) among all instances it predicted as positive. The following formula defines how the precision is calculated [29]:

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

where TP = True positive; FP = False positive

2.3.2.4 Recall

The recall metric measures the model's ability to identify all positive instances in the dataset. It is calculated as the ratio of true positive predictions to the sum of true positives and false negatives (actual positive cases that were missed by the model). For example, in ECG classification, recall signifies the model's effectiveness in capturing all instances of the target condition among the total instances that should have been identified as positive. The following formula defines how the recall is calculated [29]:

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

where TP = True positive; FN = False negative

2.3.2.5 F1-score

To assess the balance between precision and recall, the F1-score metric is utilized. This metric represents the harmonic mean between the two measures. The following formula defines how the F1-score is calculated [30]:

$$\text{F1-score} = \frac{2TP}{2TP+FP+FN}$$

where TP = True positive, FP = False positive, FN = False negative

In Figure 9, it is possible to visually perceive the relationship between the precision and recall metrics.

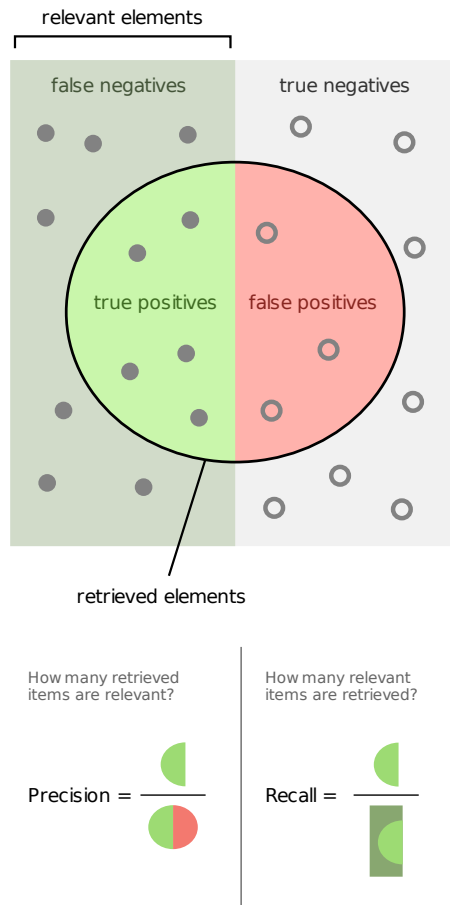


Figure 9 - Relationship between the precision and recall metrics. Image retrieved from Wikipedia [30].

3 GENERAL PLATFORM ARCHITECTURE

Developing a platform for assisting and monitoring elderly individuals poses inherent technological and physical challenges. Overcoming these obstacles is crucial to ensure the platform's effectiveness and reliability. On the technological and physical front, the integration of various medical and domotic sensors within the elderly's environment requires addressing the heterogeneity of IoT devices. To accommodate this diversity, agnostic software components are necessary to encapsulate different device types and facilitate expandability. Furthermore, some IoT devices possess limited processing capacities and utilize low-energy communication protocols like BLE, necessitating an additional gateway component to communicate with these devices and relay information to the network.

The communication aspect within this distributed system introduces its complexities. The conventional client-server paradigm is insufficient for managing data exchange among multiple IoT devices and modules. Addressing this, the Message Queuing Telemetry Transport (MQTT) protocol emerges as a lightweight and effective solution to implement the publish-subscribe paradigm [31]. Employing a centralized broker, MQTT facilitates communication by managing publishers and subscribers within specific message topics.

Data analysis forms a pivotal aspect, demanding a robust processing pipeline for the continuous assessment of diverse sensors and devices. This pipeline must accommodate various data types, from ECG and motion recognition to behavioural patterns. Depending on the analysis outcome, a sensor orchestration module triggers data collection or alerts caregivers. The adaptability of the processing pipeline to accommodate new device additions is imperative since the goal is to integrate more and more medical data types (e.g., respiration rate, oxygen levels, etc.) in the future. Given the intricacy of data patterns, particularly in medical contexts, integrating AI becomes essential to identify complex patterns and generate accurate responses beyond simple rule-based programming logic.

Likewise, the platform can effortlessly accommodate different elements like data storage, a versatile backend, caregiver dashboards, and external APIs. Tools like the TICK stack make storing sensor data easy, working well with how the platform handles data through MQTT. It's also simple to add caregiver dashboards using modern frontend tools and languages that connect with the platform over MQTT.

The platform's flexibility makes it adaptable for integration with external systems that can easily get important medical info that is produced. This adaptability also lets it work with government systems and APIs, showing how the platform facilitates the sharing of health data. With its adaptable nature, the platform is ready for future changes and can grow as new technologies and needs come up.

IoT Platform for Personal Healthcare of Elderly People

As depicted in Figure 10, the high-level architecture of the presented elderly care solution embodies this integration of devices and sensors using a gateway computer through MQTT with other software modules.

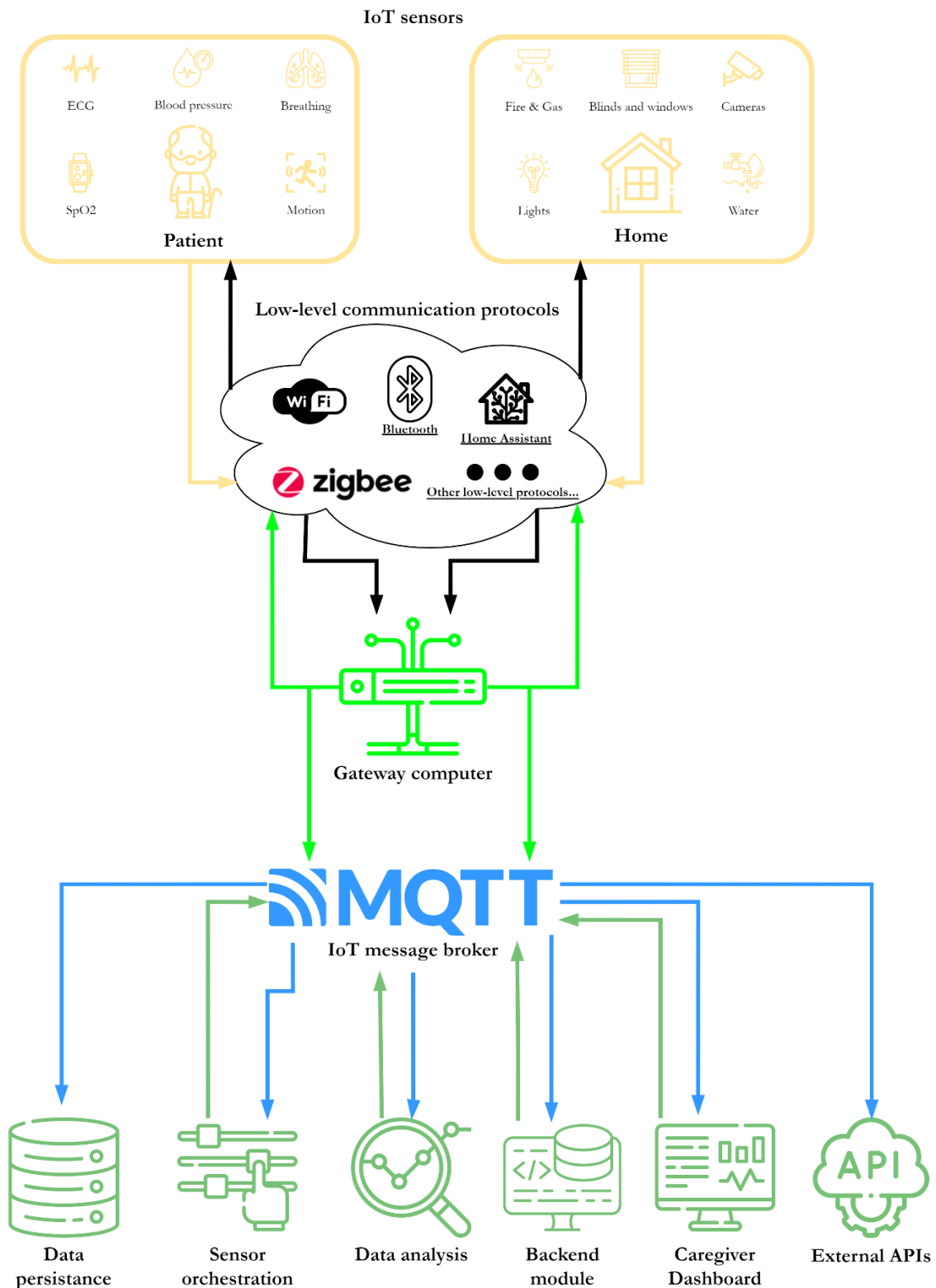


Figure 10 – General platform architecture diagram. Produced with draw.io software [32] and Flaticons [33].

The platform architecture foresees several types of software and hardware integrations that range from individual IoT sensors to more comprehensive software modules such as data analysis, sensor orchestration, and data persistence among others. However, due to limitations in terms of time and resources, a fixed number of features must be set to enable the development of a well-defined and testable proof of concept. The first feature consists of ECG Analysis (Feature 1) and plays a pivotal role by gathering data from an ECG sensor and employing Machine Learning techniques to analyse ECG normality, enabling the monitoring of the patient's cardiac health. Additionally, the feature of Motion Analysis (Feature 2) leverages data collected from motion sensors to discern the patient's physical behaviour and detect whether they are at rest or moving. Furthermore, the Sensor Simulation (Feature 3) functionality is incorporated, offering the capability to simulate physical sensors for comprehensive testing and validation purposes. The Orchestrator (Feature 4) serves as a vital software module, effectively coordinating and managing all sensors and data analysis processes. To showcase the system's capabilities, a simplified Graphical User Interface (GUI) (Feature 5) is implemented, offering a user-friendly interface for visualising and demonstrating the aforementioned features. Lastly, centralised communication over MQTT (Feature 6) underpins the entire system, facilitating efficient communication between all software components, including sensors, the Orchestrator, the GUI, and more, through a central MQTT broker utilising the publish and subscribe paradigm. These interconnected functionalities form the core of our platform, addressing the challenges and requirements of elderly care assistance and monitoring.

The technical aspects of how the aforementioned software modules are developed and how the sensor data is analysed using ML are further detailed in the subsequent chapters.

4 RELATED WORK

In recent years, the confluence of IoT, digital healthcare, and Domotics has captured considerable attention from researchers, particularly in the context of addressing the complex issues of the development of elderly care solutions. As societies worldwide undergo a demographic shift towards an increasingly elderly population, the demand for innovative solutions has become paramount. This chapter serves as a gateway to the existing body of research and developments in the field of elderly care platforms, with a specific emphasis on those harnessing the potential of IoT, digital healthcare technologies, and domotic devices.

Survey studies about IoT have been conducted since 2010 and with the new advancements, more oriented studies of IoT applications were driven in the fields of healthcare, home automation and more specifically elderly care [34]. A survey enumerates what kind of data is collected in the development of such systems. The data types are biomechanical data (e.g. the motion of the body and activities), cholesterol data, ECG data and other vital signs (e.g. blood pressure, pulse rate, respiratory rate and temperature) [35]. With the gathered data, various applications are possible such as aged care monitoring, Human Activity Recognition (HAR), clinical monitoring at hospitals, emergency conditions and more [35]. Parallel to the possibilities of IoT in elderly care contexts, a set of challenges arises. Among the various is the creation of intuitive user interfaces for the elderly, the address of security and privacy concerns and the management of the heterogeneity among IoT devices and vendors [36]. Also, the strong dependency on power sources, the context-awareness capabilities difficulties and the lack of long-term support of IoT devices for healthcare present another layer of obstacles that may present a threat to the viability of such systems [34].

In terms of practical applications of elderly care systems, many implementations are presented by the researchers in the literature. Some approaches place a strong emphasis on home automation, while others focus more on healthcare aspects. Additionally, some efforts are aiming to combine these two areas seamlessly.

4.1 Domotic and home-oriented solutions

In an Argentinian university, the authors present a specific implementation of an elderly care solution using a group of devices controlled by a central computer with dedicated software called “Sistema Integrado de Control y Automatización Asistida” [37]. The patient can control domotic devices at home using hand motions, such as turning on lights or requesting help. However, the system is hardware-intensive and may carry problems in terms of user interaction.

Portuguese researchers developed an innovative healthcare IoT system known as "We-Care" [38]. This system focuses on data collection and monitoring of the elderly

at home, utilizing a combination of hardware and software modules. The main components consist of a wearable wristband called We-Watch, service modules, and a cloud-based platform. The We-Watch wristband gathers vital data such as body and environmental temperature, pressure, light, humidity, and acceleration via built-in sensors. This data is then sent to the We-Care board, which stores it locally on an SD card or uploads it to the cloud for further analysis and storage. A notable advantage of the system is the We-Watch's impressive energy efficiency, boasting a remarkable autonomy of 12 days.

A comparable system is established in a research project where motion, temperature, and humidity sensors are installed within various rooms of the residence [39]. Due to the specific sensors it incorporates, this system solely enables the monitoring of the patient's home. It generates alerts via Twitter when irregularities occur, such as excessively high temperature, inadequate humidity levels, or instances of motion detected within the rooms, indicating the activity of the elderly individual within the house. For example, it is possible to detect if a patient spends too much time at the toilet. Moreover, the system includes a built-in idle mode, which is automatically engaged when other individuals, such as family members or caregivers, enter the premises. This feature prevents the accumulation of noise-related data resulting from gatherings within the household.

4.2 Healthcare-oriented solutions

Researchers in Malaysia developed a system whose focus is on the monitoring of heartbeat and body temperature [40]. Additionally, it provides the ability to detect falls. Furthermore, the system also integrates the functionality of door access at a patient's home. The system governs access to the patient's home door, allowing designated family members, who are entrusted with their care, to manage whether or not to grant entry to individuals such as professional caregivers.

An analogous system is implemented, by other researchers, that tracks vital signs such as temperature, heart rate, SpO₂ and fall detection [41]. The data is gathered with a Raspberry Pi and then sent to a remote InfluxDB database and to an alert system that forwards it to a mobile dashboard made in Flutter.

Other works propose integrating wearable devices with domotics and healthcare. In 2010, SmartVest was presented which consists of a domotic wearable vest that combines clothing with healthcare and house control in a user-friendly way. However, challenges arise in the textile and technology industry to create devices that are unnoticeable when integrated into clothing [42]. In 2019, the authors suggest a wearable domotic solution using medical sensors like ECG, SpO₂, and temperature, connected to a central processing unit that communicates with the home domotic network through MQTT to automate tasks [43].

A more precise investigation centres on an advanced care system that incorporates an ECG sensor that transmits signals via Bluetooth to a central server for signal

processing [44]. The sensor is integrated into a wearable shirt, designed for convenient wearing with a zipper. Furthermore, it facilitates waveform visualization through an Android application. A noteworthy aspect of this research is its capacity to analyze intricate data, specifically ECG, which distinguishes it from other studies concentrating primarily on simpler metrics like heart rate, temperature, and SpO₂.

5 ECG ANALYSIS

5.1 Fundamentals of Electrocardiography

The field of electrocardiography involves the study of the electrical activity of the heart through an ECG that captures the depolarisation and repolarisation of atrial and ventricular cells over time [45]. Given the fact that the body can represent a giant conductor, by placing electrodes on the skin's surface, the voltage of the heart's electrical field is detected and an ECG reading can be produced. The resulting ECG graph displays voltage (in millivolts, mV) plotted against time (in seconds). To minimize voltage fluctuations caused by skeletal muscle movement, ECGs are typically recorded with the patient at rest. The ECG has evolved into a powerful diagnostic tool for heart disease, especially for the detection of arrhythmias and acute myocardial infarction. The use of the ECG has become a standard of care in cardiology, and new advances using this technology are continually introduced [45].

5.1.1 ECG curve

The pattern of the ECG curve is characterized by the regular and frequent beating process of the heart, which results in a rhythmic waveform [45]. The voltage measurements obtained from the ECG recording follow a distinct pattern that corresponds to the electrical activity of the heart during each heartbeat. This pattern typically consists of a series of identifiable waves, including the P wave, Q wave, R wave, S wave and T wave. The Q, R and S waves are also known as the QRS complex. These waves represent specific electrical events in the heart, such as atrial and ventricular depolarization and repolarization. The repeated occurrence of these waves in a consistent and organized manner reflects the regularity and synchrony of the heart's contractions, providing valuable information about the heart's functioning and overall cardiac health [45].

In Figure 11, an example of a normal ECG recording is presented and in Figure 12, the position of the different waves is illustrated.

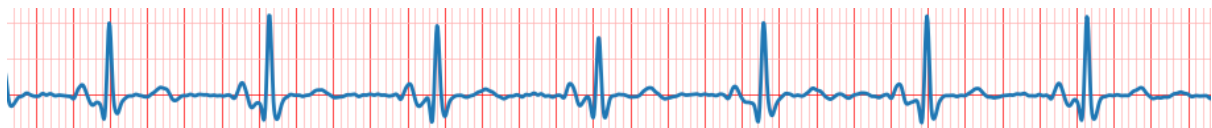


Figure 11 – Random normal 10-second ECG recording from the DeepFake database [46]. Chart generated with Python's "ecg-plot" library.

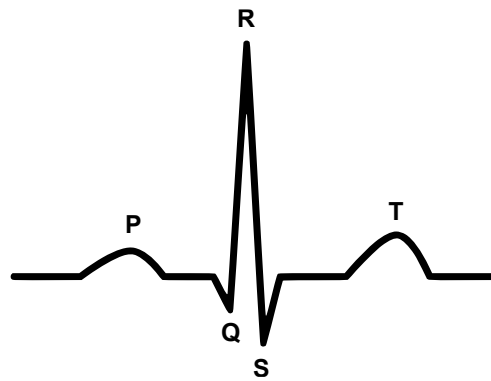


Figure 12 – Characterization of an ECG heartbeat segment with the five labelled waves from left to right (S, QRS complex and T). Image retrieved from a Wikipedia article about electrocardiography [47]

5.1.2 Lead system

The ECG recordings are performed by installing a couple of electrodes in the human body which capture the electrical activity of the heart. However, the way the electrodes are placed in the body generates different types of ECG waveforms. To this extent, several lead systems were defined to uniformize how electrodes are positioned in the body. An ECG lead is a specific electrical pathway between two electrodes. The three lead positions that are most frequently used today are known as leads I, II, and III. To understand the placement of these leads, the body's torso can be interpreted as an equilateral triangle. This configuration is called Einthoven's triangle. In Figure 13, the Einthoven triangle can be illustrated for the three limb leads [45].

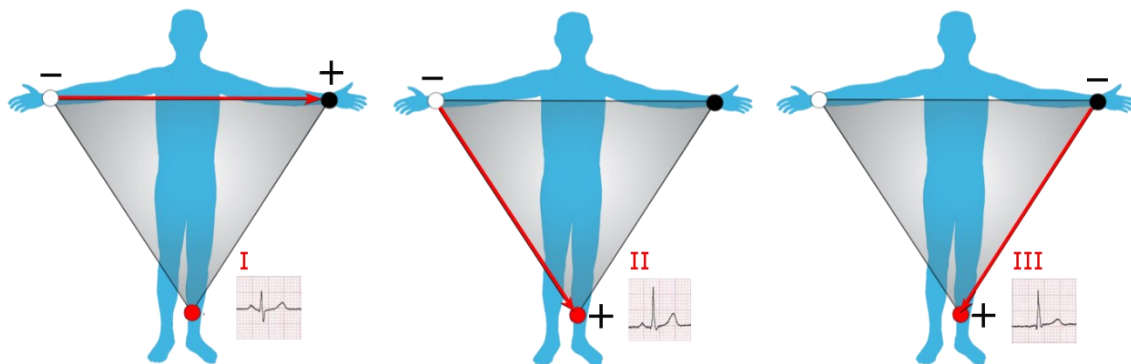


Figure 13 – Representation of the three limb leads (I, II and III) and their corresponding ECG waveforms following the Einthoven triangle. Image retrieved from a Wikipedia article about electrocardiography [47]

The transition to the 12-lead ECG further expanded the diagnostic capabilities by incorporating additional leads on top of the first three leads (I, II and III) and capturing electrical signals from different angles, enhancing the interpretation of cardiac abnormalities [45].

5.1.3 ECG devices

Various ECG devices are available on the market, ranging from those capable of recording multiple leads, such as the standard 12-lead ECG used in hospitals for its precision in medical diagnostics, to more user-friendly IoT-based devices that offer a comfortable experience but provide fewer leads. The latter allows for the development of healthcare applications, as exemplified in this project. However, it's important to note that IoT-based ECG device development is still in its emergent stages, with limited availability and relatively high costs. This project primarily focuses on lead I ECG due to constraints related to the project timeline and financial resources.

5.2 Machine Learning for ECG analysis

In the hospital, ECG analysis is traditionally carried out by professional cardiologists who meticulously examine electrocardiogram recordings of the patients. These experts inspect the waveform patterns, durations, and intervals to identify potential cardiac abnormalities. However, this process can be time-consuming and subject to variations in interpretation. Moreover, in the context of this elderly care platform, it is unfeasible to hire professionals who continuously analyse the ECG data of every patient.

In this subchapter, the application of ML techniques is explored to automate and enhance this analysis process, with the ultimate goal of assisting healthcare professionals and improving the accuracy and efficiency of ECG interpretation.

The analysis of ECG using ML techniques involves several steps such as the discovery of datasets or the gathering of training data from patients directly, the preprocessing of the gathered data, the extraction of features and the training of ML models. The following subchapters describe how each one of the mentioned concerns is addressed in the context of this elderly care platform.

5.2.1 Related work

The usage of Machine Learning to analyse ECG data is studied by many researchers. Experiences with different datasets, ML models, pre-processing techniques and feature extraction methods are carried out to converge into an optimal solution concerning ECG analysis and abnormalities detection.

Many studies use the well-known MIT-BIH Arrhythmia Database [48] as the training input for the ML models that are tested. The database consists of 48 half-hour recordings from distinct subjects and the recordings are labelled by professionals to define what type of abnormality is present. A survey of 2015 on the classification of ECG signals using ML techniques [49] presents several types of research using the MIT-BIH database. In [50], the performance of Support Vector-Machines (SVM) and ANNs is compared. The authors in [51] propose a neural network that achieves

over 90% accuracy. Also, in [52], neural networks are used to test the performance of heart arrhythmia detection.

Another analogous survey of 2018 on ECG analysis [53] enumerates several studies that use different ML models, pre-processing and feature extraction techniques, and databases. Some suggested models are Neural Networks, the KNN, SVMs, Bayesian Networks, and Decision Trees, among others. Also, many databases are mentioned such as the MIT-BIH Arrhythmia Database [48] and the PTB Diagnostic ECG Database [54], which are the most cited and used.

In a study in 2020, researchers conducted a large-scale study on ECG classification using DNNs with residual network architecture [55]. They gathered a very large dataset representing more than 2.3 million recordings of more than 1.6 million people with 10-second long recordings [56]. The goal was to achieve an optimal ECG classification capacity without any feature engineering and using 12-lead ECG, which means that the ECG recordings are fed raw to the neural network as input. Their results consisted of metrics (precision, recall and F1-score) of more than 90% trying to predict 6 different types of ECG abnormalities.

In 2022, a study was proposed to classify ECG recordings using Explainable Boosting Machines [57]. In opposition to [55], the goal is to isolate every heartbeat of the recordings and process with feature extraction for every beat using calculations based on the P and T waves and the QRS complex. As a result, an accuracy of over 95% was achieved in the evaluation of heartbeats. The MIT-BIH database was used as training and test data.

5.2.2 Dataset discovery

The process of finding datasets, which act as training data for the ML models used in ECG analysis, involves exploring various datasets mentioned in literature and utilized by related studies. In this regard, an incremental approach is adopted to gather the datasets individually, one at a time. In the context of this work, the goal is to gather datasets that have information about normal and abnormal heart recordings. The exact heart condition description of abnormal recordings is not relevant at this development stage. Since this project handles only a lead I ECG device, the dataset needs to have information about the ECG recordings of the given lead.

To this extent, two criteria are defined to decide whether the found datasets are eligible or not to be used. The first is having data about ECG recordings of lead I and the second is having data that has been labelled by a professional in whether the recording is normal or abnormal.

Although it is the most frequently used dataset, the MIT-BIH Arrhythmia Database is not eligible for the sake of this study because it doesn't contain ECG recordings of lead I.

Table 1 lists the datasets that were discovered and used in this project.

IoT Platform for Personal Healthcare of Elderly People

Table 1 – List of datasets used in this project for ECG analysis

Name	Sample size in seconds (s)	Frequency in Hertz (Hz)	Leads	Total Used Quantity in seconds (s)	Heart conditions in seconds (s)
CODE-15%: a large-scale annotated dataset of 12-lead ECGs [56]	10 s	400 Hz	All 12 leads	993030 s	382190 s of Normal 610840 s of Abnormal
A large-scale 12-lead electrocardiogram database for arrhythmia study [58]	10 s	500 Hz	All 12 leads	451450 s	0 s of Normal 451450 s of Abnormal
PTB Diagnostic ECG Database [54]	Between 32 s and 120 s	1000 Hz	All 12 leads and 3 Frank XYZ leads	55350 s	9280 s Normal 46070 s of Abnormal
DeepFake electrocardiograms using generative adversarial networks are the beginning of the end for privacy issues in medicine [46]	10 s	500 Hz	All 12 leads	250000 s	250000 s of Normal 0 s of Abnormal
PTB-XL, a large publicly available electrocardiography dataset [59]	10 s	500 Hz	All 12 leads	217990 s	71720 s of Normal 146270 s of Abnormal

All gathered datasets, apart from CODE 15% and DeepFake, were downloaded from PhysioNet which is a collection of openly accessible medical research data, overseen by the MIT Laboratory for Computational Physiology in the United States of America [60]. As an exception, the DeepFake and CODE 15% datasets were downloaded from their independent repositories created by their respective authors [46], [56].

The combined recording duration of the five datasets comprises a total of 1967820 seconds of ECG recordings. Among these recordings, 713190 seconds represent normal recordings, while the remaining 1254630 seconds encompass abnormal recordings. In other words, the normal recordings represent more or less 36 % of all the recordings and the abnormal recordings represent more or less 63 %. The percentages reveal an imbalance in the final dataset intended for use. So, it remains uncertain whether this class imbalance will pose an issue when training the models later. Therefore, further analysis on the imbalance topic will be conducted for comparison purposes and it will be discussed whether it has or not an impact.

5.2.3 Pre-processing

5.2.3.1 Dataset uniformization

There are some persistence format variations among all downloaded datasets. The ones available in the PhysioNet repository are saved in the WaveForm DataBase (WFDB) format which can be read using PhysioNet's WFDB Software Package [61]. The DeepFake dataset is stored in simple text files that can be parsed without complexity using traditional file reading methods. Lastly, the CODE 15% dataset is stored in the Hierarchical Data Format 5 (HDF5) file format which can be parsed using the official HDF5 library [62].

After parsing all five datasets and as a way of easing the processing and concatenation of the different datasets, several steps of uniformization are performed. To them belong the downsampling into a common frequency using the Fourier transform, the scaling of the signal to a specific range, the signal cleaning and filtering to remove noise, and the extraction of features. In the end, every uniformized dataset is stored in homogeneous Comma-Separated Values (CSV) files.

Figure 14 illustrates how each step of uniformisation for all datasets is performed. In the subsequent sections, a more detailed explanation is given about the specificities of each uniformisation procedure.

The final CSV files are going to be made available at <https://www.kaggle.com/datasets/a21270652/lead-i-ecg-data-raw-and-calculated-features>.

IoT Platform for Personal Healthcare of Elderly People

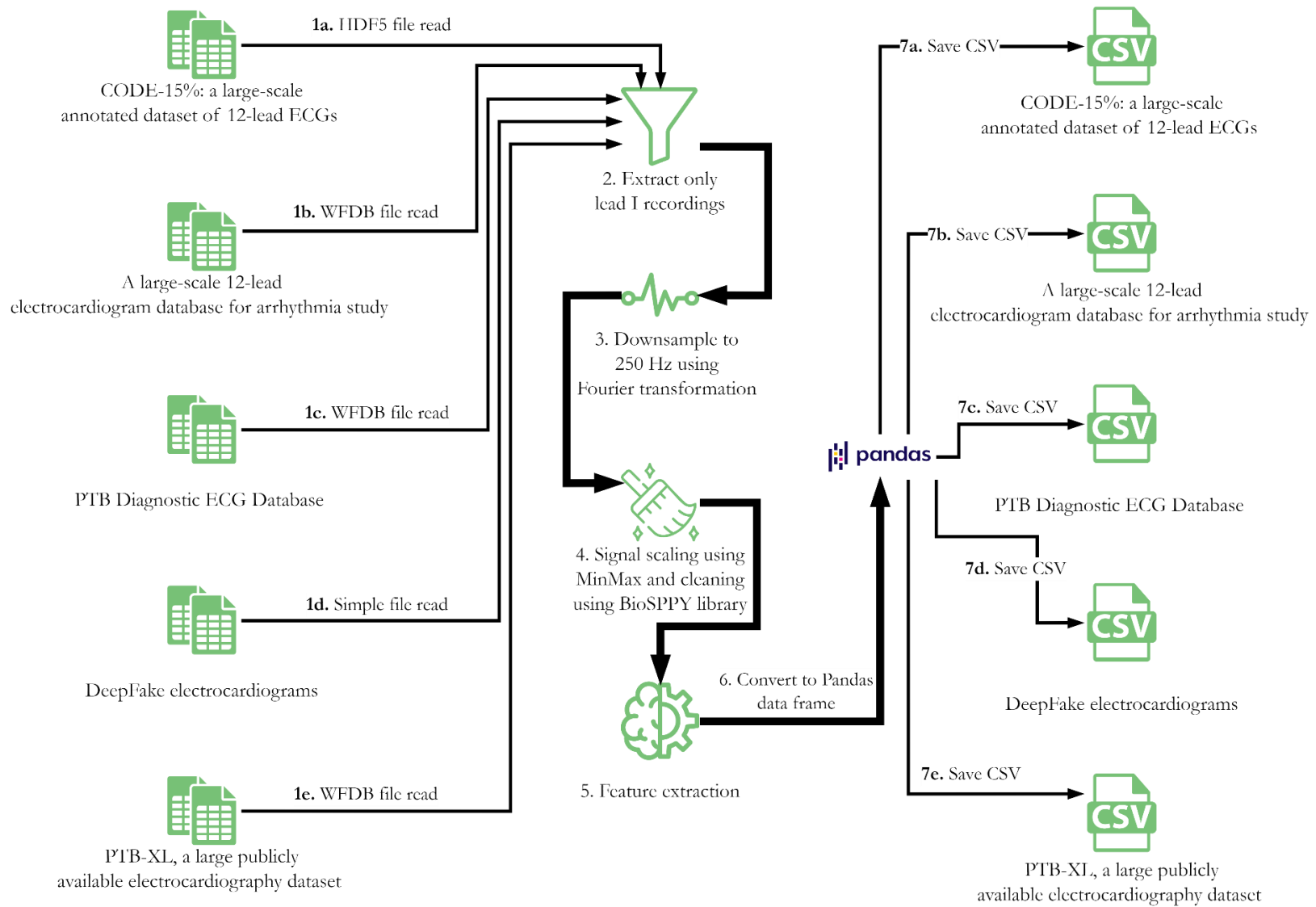


Figure 14 – Diagram explaining how the uniformisation for all datasets is performed. Produced with draw.io software [32] and Flaticons [33].

5.2.3.2 Downsampling

Due to limitations in terms of processing power availability, a downsampling procedure is applied to all datasets. That way, the training times of the models and the amount of random access memory (RAM) that is needed can be optimized. A study demonstrates that ECG signals recorded above 120 Hz achieve enough accuracy to be analysed [63]. Also, in a survey on ECG analysis, several works are cited where downsampling to 250 Hz was done due to computational cost [53]. Nevertheless, downsampling data to 250 Hz carries the risk of losing information that is only detectable at the original frequency. The datasets employed here span frequencies from 400 Hz to 1000 Hz, implying that the downsampling process could result in the loss of up to 75% of the data. While numerous studies affirm the sufficiency of accuracy above 120 Hz, it is imperative to consistently consider this trade-off and its impact on the analysis of the results.

To this extent, a threshold of 250 Hz is settled for the common frequency for all datasets. To downsample the data, the “resample” function in the “signal” module of the SciPy library is used [64]. The function downsamples the data using the Fourier transform method.

5.2.3.3 Signal scaling and cleaning

After converting all datasets into the same frequency baseline (250 Hz), the ECG signal needs to be scaled and cleaned. Firstly, the data needs to be scaled to a specific range since the signal amplitude may vary from individual to individual because of the distinct body morphology of every patient and the type of recording ECG device that was used [45]. Factors like the body weight or small unexpected motions of the patient may alter the amplitude of the signal, which can bias results [45]. By using the “MinMaxScaler” class from the “preprocessing” module in the Scikit-learn library, it is possible to scale the signal to confine the amplitude within the range of [0;1].

The second step is to clean the signal using the BioSPPy library [65], more specifically the “ecg” function from the “ecg” module. This function passes the ECG signal through a filter that removes unwanted noise from it. It also detects the R peaks present in the ECG sample which are useful later in this chapter.

5.2.4 Feature extraction

In pursuit of harvesting as much valuable information from the datasets as possible, some feature engineering is applied to the pre-processed data. There are several possible approaches regarding several aspects of feature extraction. The first aspect is the time window, which can vary from individual heartbeat chunks [57] to several seconds or minutes [55]. The second aspect is whether the feature vector represents the raw ECG signal out-of-the-box or a set of calculated features based on the P, QRS complex and T waves. Also, a combination of the two approaches is possible.

5.2.4.1 P, QRS complex and T waves

The synchronous behaviour of the ECG curve produces the characteristic P, QRS complex and T waves (Figure 12). Some works [57] make use of the waves to calculate important features that help describe the ECG curve. Relevant features can be the amplitude difference between two opposite peaks (e.g. R and S) or the time difference between two different peaks (e.g. R and P, or R and T).

Since the downloaded datasets are organized in time windows of 10 seconds or more, it is necessary to isolate the individual heartbeats and detect the different P, QRS complex and T waves. To help with this, the function that is used to filter the ECG signal from the BioSPPY module is availed because it also retrieves the locations of the different R peaks in the signal. The R peak locations indicate the presence of a new heartbeat. Based on the R peaks, it is possible to detect the other P, Q, S and T waves. To achieve it, the function “ecg_delineate” from the NeuroKit2 library is used to extract the remaining waves. It takes the input of the raw signal and the previously retrieved R peak locations to detect the other remaining four peaks (P, Q, S and T).

To this extent and inspired by a previous study [57], the following calculations are made for all heartbeats:

- a) R and S amplitude difference;
- b) R and P time difference;
- c) R and T time difference;
- d) time difference between two consecutive R peaks.

When calculating the mentioned features on a time window with multiple heartbeats, each feature is calculated for each heartbeat of the time window and then the arithmetic mean is calculated to get an average value for each feature inside the time window.

5.2.4.2 Creation of feature vectors

As a way of testing the different approaches of time windows and the usage of raw ECG signal data and/or calculated ECG wave features, five different feature vectors are defined. The goal is to test the different feature vectors and see which ones have better performance when fed to ML models.

In Table 2, the five methods to create the feature vectors are listed. Methods 1, 2 and 3 are based on a 10-second time window, where Method 1 only contains the raw ECG data, Method 2 contains the raw ECG data and the respective calculated features of the time window and Method 3 only contains the calculated features for the time window. Methods 4 and 5 are based on individual heartbeats, where Method 4 has the raw signal of the heartbeat with its calculated features and Method 5 only contains the calculated features.

In this context, a feature can denote either a specific ECG value at a particular time point or a calculated value based on the P, QRS, and T waves. For instance, with the ECG recording frequency set at 250 Hz, a 10-second recording yields 2500 values. Consequently, Method 1 comprises 2500 features, each representing an ECG value within that 10-second interval. Method 2 includes the 2500 raw ECG values along with 4 feature calculations derived from the waves. Method 3 solely considers the 4 feature calculations. Similar reasoning applies to methods 4 and 5, but with the distinction that only one individual heartbeat is considered instead of a 10-second recording, resulting in fewer raw features.

Table 2 – List of the five feature vector creation methods

Name	Time window	Raw ECG data	Calculated ECG features	Number of features for training
Method 1	10 seconds (multiple heartbeats)	Yes	No	2500 features (raw ECG only)
Method 2	10 seconds (multiple heartbeats)	Yes	Yes	2504 features (2500 raw ECG features and 4 calculated features)
Method 3	10 seconds (multiple heartbeats)	No	Yes	4 calculated features
Method 4	Individual heartbeat	Yes	Yes	153 features (150 raw ECG features and 3 calculated features)
Method 5	Individual heartbeat	No	Yes	3 calculated features

5.2.5 Model training

In the quest for exploring the different available ML models that could be used for ECG classification, an empirical approach is driven to compare the different types of models systematically. Since the gathered datasets contain only two distinct labels, namely “normal” and “abnormal”, the kind of problem that is analysed consists of binary classification. The goal is to conduct a combinatory analysis where each ML model is tested with each feature vector in Table 2. Furthermore, to complete the analysis, each model and feature vector is tested with balanced and unbalanced data to assess how the imbalance of the training dataset affects performance.

Three traditional models are tested, such as the Random Forest classifier, the k-Nearest Neighbours classifier and the Explainable Boosting Machine.

The Random Forest is employed in several studies for ECG classification with the MIT-BIH database where it achieves an accuracy of over 95% [53], [66]. Other studies employ the KNN for ECG classification purposes using the MIT-BIH database achieving sensitivity and specificity of over 90% [53], [67]. Also, the usage of EBMs is inspired by an analogous study where the MIT-BIH database is used for training and an accuracy of over 95% is achieved.

To diversify the approach, three DNNs are employed and tested, with one based on residual units with convolutional layers and the other two only containing convolutional layers without residual units. The first DNN is inspired by the study from Minas Gerais mentioned before [55], where a DNN is created using convolutional layers and residual units. In the previous study, the Neural Network performed well in diagnosing six heart conditions with precision, recall and specificity of over 85%, 90% and 95% respectively. However, the neural network is built for a 12-lead ECG input recorded at 400 Hz for 10 seconds. Also, the output layers were set to classify six different classes. For this study, small modifications are made to adapt the DNN to lead I ECG downsampled at 250 Hz and for binary classification of normal and abnormal conditions.

The second neural network is inspired by the previous one with fewer layers and less complexity. The goal is to try to create a simpler DNN that is not as resource-consuming in terms of training and predicting time as the original Minas Gerais DNN. By also using convolutional layers like in the original DNN combined with dropout and maximum pooling layers to avoid overfitting, a faster regularized neural network is developed. However, the adaptations result in a trade-off where the faster training times slightly compromise the overall metrics.

The third neural network is the same as the second neural network but without regularization. In other words, it hasn't any layers to avoid overfitting as a way of testing the effects of those layers.

Table 3 summarizes which classifiers are submitted to test in this study.

Table 3 – List of classifiers submitted to test in this study

Classifier name	Type	Description
Random Forest	Traditional model	Random Forest classifier with all default parameters
KNN	Traditional model	KNN classifier with all default parameters
EBM	Traditional model	EBM classifier with all default parameters
Minas Gerais Neural Network	Neural Network	Neural Network repurposed from another study [55]. The architecture is presented in Appendix D. The original Python code can be found in [55].
Exploratory Neural Network with Regularization	Neural Network	A simplified version of the Minas Gerais Neural Network without residual units. The architecture and Python code are presented in Appendix E.
Exploratory Neural Network without Regularization	Neural Network	Exploratory Neural Network without overfit-combatting layers (Dropout and MaxPooling). The architecture and Python code are presented in Appendix E.

5.2.5.1 Training and testing environment

In this study, Python [68] is used with the Jupyter Notebook environment [69] to pre-process the datasets, generate the feature vectors, and create and train the ML models.

The “RandomForestClassifier” class from the “ensemble” module in the Scikit-learn library is used to represent the Random Forests [70]. The “KNeighborsClassifier” class from the “neighbours” module in the Scikit-learn library is used to represent the k-Nearest Neighbour [70]. The “ExplainableBoostingClassifier” class from the “glassbox” module in the InterpretML library represents the EBM [18]. For the creation of the neural networks, the TensorFlow library is used [71].

To the extent of homogenizing the training and testing environment for all models and feature vector combinations, several procedures are done to avoid biases from different training and testing instances.

Firstly, the random seed is set to an arbitrary value of 42 in the Python and TensorFlow environments. Also, the same value is passed as the random state argument for reading the datasets and executing the splitting of the training and test data. Additionally, when executing the training, the random state of 42 is also passed as an argument.

The splitting of the train and test data is set to a ratio of 80% and 20% respectively. Exceptionally, to train the neural networks, an additional split of 10% of the training data is made for the validation data.

In terms of hyperparameters, all models are initialized with their default values and no hyperparameter tuning or grid search analysis is conducted.

In addition, all testing combinations of ML models and feature vectors are performed using a balanced and unbalanced version of the dataset as input. That way, it is possible to compare the impact of balancing the dataset or not. It can be notorious since more abnormal ECG samples than normal ones are available.

5.2.6 Results and discussion

In this study, multiple ML models undergo an evaluation process in conjunction with various feature extraction techniques to ascertain the most suitable choice for this project. The evaluated results consist of predictions on test instances that were not encountered during model training, ensuring that the models' eventual high performance is not attributed to overfitting to training data. Additionally, an analysis is conducted to determine the impact on the performance of utilizing a balanced dataset versus an unbalanced one. In total, 60 test instances are conducted, encompassing each ML model combined with each feature extraction method, along with the consideration of a balanced or unbalanced dataset. Appendix B and Appendix C contain performance charts for each conducted test, facilitating straightforward comparisons between individual models and feature extraction methods. This is achieved through two distinct types of comparison charts. Within Appendix B, comparisons are made between individual feature extraction methods within a specific ML model. Conversely, in Appendix C, comparisons are drawn between individual ML models using a specific feature extraction method.

To evaluate the influence of dataset balance, all charts are duplicated for each balancing scenario. This duplication allows visual examination of tests conducted with both balanced and unbalanced datasets.

The following sections aim to highlight the main responses obtained by the results from the tests. All observations are based on the charts that are available in the appendices (Appendix B and C).

5.2.6.1 Class balance

The dataset used to feed the ML models in this analysis suffers from class imbalance since approximately 36% of the instances represent normal recordings and 64% represent abnormal recordings. In Machine Learning, the issue of class imbalance presents a persistent challenge for classifiers [72], rendering them susceptible to overfitting on the prevalent class. Also, when a dataset is unbalanced, the sole usage of the accuracy metric may not reflect the reality of all classes which can be misleading. In those cases, it is important to also analyse precision and recall.

Through empirical comparison of all test instances across the various ML models and feature extraction methods, the findings indicate that balancing the dataset does not seem to yield a noticeable enhancement in the overall metrics. Typically, when confronted with unbalanced datasets, the accuracy is often higher with a greater disparity between precision and recall. Nevertheless, in this scenario, the accuracy of the unbalanced dataset surpasses that of the balanced dataset test, while concurrently demonstrating elevated precision and recall metrics.

Henceforth, considering the superior performance exhibited by the original and unbalanced training dataset compared to the balanced version, all subsequent observations are directed towards the unbalanced dataset tests.

5.2.6.2 Performance of the classifiers

Due to the extensive volume of results obtained in this study, it is impractical to present them all within a table. Table 4 provides a concise summary of key comparative findings. While there may be additional noteworthy results, we have prioritized discussion of the most significant scores to maintain analytical clarity. The extended results findings with all scores can be found in the appendices, namely Appendix B and Appendix C.

IoT Platform for Personal Healthcare of Elderly People

Table 4 – Summary of key comparative results (unbalanced dataset only)

Model	Feature extraction method	Accuracy	Precision	Recall	F1-score
Random Forest	Method 1	74 %	72 %	95 %	82 %
	Method 3	71 %	77 %	78 %	77 %
	Method 4	79 %	83 %	86 %	84 %
KNN	Method 1	65 %	68 %	86 %	76 %
	Method 3	68 %	75 %	76 %	75 %
	Method 4	72 %	80 %	78 %	79 %
EBM	Method 1	74 %	75 %	88 %	81 %
	Method 3	72 %	79 %	78 %	78 %
Minas Gerais Neural Network	Method 1	80 %	87 %	82 %	84 %
	Method 3	71 %	77 %	79 %	78 %
	Method 4	82 %	78 %	97 %	86 %
Exploratory Neural Network with Regularization	Method 1	80 %	87 %	80 %	83 %
	Method 3	65 %	68 %	86 %	76 %
	Method 4	76 %	80 %	85 %	82 %
Exploratory Neural Network without Regularization	Method 1	75 %	78 %	84 %	81 %
	Method 3	71 %	77 %	77 %	77 %
	Method 4	79 %	83 %	86 %	84 %

5.2.6.2.1 Traditional models' performance

The traditional Random Forest and KNN classifiers show the worst performance among all tests, especially when submitted to the feature extraction methods of raw ECG inputs (Method 1). Their accuracy peaks between 70% and 75%. However, the two classifiers show high recall of 95% and 84% respectively when analysing raw ECG, but subsequently also have a high precision trade-off of 72% and 68%

respectively. The only exception is when the previous two classifiers are submitted to the heartbeat features and raw signal method (Method 4) where they achieve higher accuracy of 79% and 72% and an F1-score of 84% and 79% with an optimal balance between precision and recall. In this method, they even compete with the other tested classifiers in terms of performance.

The EBM classifier shares similar performance with the Random Forest and the KNN classifiers with small improvements among all the metrics. For example, in raw ECG input (Method 1), the EBM achieves an accuracy of 74% with an F1-score of 81% while assuring better precision and recall balance than the other two classifiers. Unfortunately, due to unexpected technical encounters, the EBM classifier could not be trained with the input of feature extraction methods 4 and 5. To this extent, no results are available for this case and no comments can be drawn on them.

5.2.6.2.2 Deep Neural Networks

The trained DNNs exhibit good performance, particularly when exposed to raw ECG input (like Method 1), in contrast to the performance of the traditional models in this scenario. The adapted Minas Gerais Neural Network attains an accuracy of 80%, along with precision, recall, and F1-score values of 87%, 82%, and 84% respectively. Similarly, the Exploratory Neural Network demonstrates comparable results, achieving accuracy, precision, recall, and F1-score of 80%, 87%, 80%, and 83% respectively. Conversely, the unregularized version of the Exploratory Neural Network, without overfit-combatting layers like Dropout or MaxPooling, yields scores of 75%, 78%, 84%, and 81% respectively. Here, it is possible to conclude at first glance that neural networks exhibit greater proficiency in uncovering concealed patterns within raw ECG data compared to traditional models. Moreover, it is observable that the overfit-combatting layers applied to the regularized Exploratory Neural Network lead to a marginal improvement in its overall performance scores.

On the other hand, the performance of neural networks applied to feature-only input methods like Method 3 and Method 5 drops significantly. For example, in this case, the Exploratory Neural Network reaches accuracy, precision, recall and F1-score of 65%, 68%, 86% and 76% respectively. The other two DNNs follow the same tendency in terms of performance degradation.

Analogously to the traditional methods, when submitted to the heartbeat features and raw signal method (Method 4), the three neural networks perform well too. For example, the Exploratory Neural Network reaches accuracy, precision, recall and F1-score of 76%, 80%, 85% and 82% respectively. However, in the unregularized version of the Exploratory Neural Network where no overfit-combatting layers exist, the performance concludes to be higher than the previous one with scores of 79%, 83%, 86% and 84% respectively.

5.2.6.3 Feature extraction methods' performance

This analysis reveals noticeable variations in the performance of the five feature extraction methods contingent upon the model employed for their training. Specifically, within the context of neural networks, the raw-type methods (Method 1 and Method 2) exhibit superior performance, as earlier mentioned. Conversely, feature extraction methods inclined towards feature engineering, such as Method 3 and Method 5, tend to be better when implemented with traditional ML models. Notably, feature extraction method 4, encompassing both individual heartbeat features and the raw heartbeat data, attains commendable performance across all classifier types, encompassing both traditional models and neural networks.

5.2.6.4 Final observations

This analysis permits us to understand which ML models work best with which feature extraction methods. Although no universal answer can be drawn of which is the best classifier or the best feature extraction method, it is still possible to choose the best classifier depending on the needs and problem restrictions. As an example, the Neural Networks in general demonstrate here that they are capable of uncovering the hidden patterns of ECG and permit quite satisfactory classification performance. Nonetheless, Neural Networks also have the disadvantage of having low interpretability when compared to more traditional classifiers due to their vast hidden layers structure. This may be problematic when dealing with medical information where there is a need to interpret how a given classifier has arrived at a certain decision. To this extent, the usage of the feature extraction method that is based on heartbeat features and raw heartbeat data (Method 4) may be a more favourable choice since the traditional methods such as Random Forest proved to be capable of making good predictions. This way, it is possible to have a reasonable performance and still maintain the high interpretability of the models.

Another important observation is the fact that none of the trained classifiers achieves the scores of other related studies mentioned previously [50], [52]. However, most of those studies have in common the fact that they use only one dataset, namely the MIT-BIH Arrhythmia [48]. Despite its widespread adoption in state-of-the-art research, this dataset is marked by certain limitations. These include the absence of lead I recordings and its constrained scope, confined to data collected from only 48 distinct patients. The dataset's relatively modest size could be a contributing factor enabling the utilized classifiers to promptly discern ECG patterns and achieve higher scores. In contrast, our study revolves around an extensive dataset combined from five distinct sources in the literature. The composition of this dataset potentially introduces heightened complexity to the underlying data patterns. This complexity arises from the inclusion of hundreds of thousands of recordings encompassing diverse demographics, pathologies and recording devices. Despite this diversity, the dataset uniformly employs lead I as the recording lead,

thereby fostering consistency in our analyses. Therefore, the classifiers used in this study may not be fine-tuned enough to the vast training dataset to achieve more optimal performances similar to other studies.

To conclude this discussion, it is noteworthy to mention the good performance exhibited by the Minas Gerais Neural Network in ECG classification. In the original study [34], the authors engineered a Neural Network capable of classifying 12-lead ECG recordings. The training dataset comprised a substantial collection gathered by the researchers themselves. For our study, the same neural network is repurposed, adapted to solely process lead I recordings and operate at a lower frequency of 250 Hz. Remarkably, despite being trained with significantly fewer instances than the original configuration and without the presence of data from the other 11 leads, it attains an accuracy of 80%. Further, it achieves precision, recall, and F1-score values of 87%, 82%, and 84% respectively, when compared to the original study's metrics of 92%, 93%, and 93% for precision, recall, and F1-score respectively. However, it's noteworthy that the Minas Gerais Neural Network used in this study possesses the drawback of demanding substantial computational resources. In contrast, the more simple regularized Exploratory Neural Network, derived from the architecture of the original Minas Gerais Neural Network, achieves remarkably similar outcomes while significantly reducing both training and prediction times. This adaptation proves particularly suitable for the IoT-oriented nature of this project, where energy efficiency constitutes a pivotal consideration.

6 MOTION ANALYSIS

In this project, the concept of motion analysis refers to the study and analysis of body movements exhibited by a patient. By examining how a patient moves, valuable insights can be gained regarding their physical condition and activity levels. This information serves as a complementary source to the analysis of ECG data. The analysis of motion helps determine whether a patient is at rest or in motion, allowing the detection of optimal moments to record ECG readings without interference from human movement. Furthermore, motion analysis provides a broader understanding of the patient's overall physical activity, shedding light on factors such as walking frequency, prolonged periods of rest, and patterns of motion. By delving into the idea of motion analysis, a more comprehensive view of the patient's level of activity can be attained.

This project makes use of a smartphone as the primary source for motion data, given the absence of viable alternatives such as dedicated sensors or devices for this proof of concept. However, using the smartphone provides practicality and accessibility in capturing diverse motion data. The widespread availability and built-in sensors facilitate reliable and convenient data acquisition.

6.1 Motion sensors

Accelerometer and gyroscope sensors play a fundamental role in capturing and measuring the concept of motion or movement. Additionally, the GPS permits the precise geolocalization of an object and the tracking of its movement. These sensors are integrated into devices such as smartphones, fitness trackers, and wearable technologies, allowing for precise monitoring and analysis of body movements.

By utilising the data from the accelerometer and gyroscope sensors, the motion and movement can be quantified and analysed in a precise and objective manner. These sensors serve as essential tools for capturing, recording, and interpreting the intricate dynamics of human motion, enabling applications in various fields such as sports performance analysis, rehabilitation monitoring, and activity tracking for overall health and well-being.

6.1.1 Accelerometer

The accelerometer sensor measures the acceleration forces acting on an object or device in three axes: X, Y, and Z. It detects changes in velocity, enabling the measurement of linear motion, such as walking, running, or even the slightest tilts or jolts. By analysing the data from the accelerometer, the intensity, duration, and frequency of movements can be determined, providing valuable insights into the level of activity and physical exertion.

6.1.2 Gyroscope

On the other hand, the gyroscope sensor measures the rotational or angular velocity of an object or device around its three axes: X, Y and Z. It tracks the orientation and angular changes, allowing for the detection of rotations, twists, or turns in the body's movement. The gyroscope sensor provides information on the direction, speed, and angular changes of motion, enhancing the understanding of complex movements and capturing nuances in physical activities.

6.1.3 GPS

The GPS is a vital technology that contributes to the understanding of spatial positioning and movement. By leveraging a network of satellites, GPS receivers can accurately determine their precise location on Earth. The GPS measures the time it takes for signals to travel from multiple satellites to the receiver, enabling triangulation and calculation of the receiver's coordinates. With the help of GPS, movements, trajectories, and distances travelled can be tracked, facilitating navigation, mapping, and location-based applications.

6.2 Machine Learning for motion analysis

The analysis of motion data can present significant complexities when approached through traditional conditional logic methods. The multidimensionality of motion data, characterized by the involvement of multiple sensors and the abundance of variables associated with each sensor, introduces intricacies that can be difficult to navigate using conventional analytical approaches. The volume and diversity of data generated by these sensors need more advanced techniques, such as ML algorithms, to effectively uncover patterns, relationships, and insights within the intricate fabric of motion data. Analogously to ECG analysis, using advanced methodologies, the challenges posed by the complexity of motion data can be overcome and its full potential for meaningful analysis and interpretation can be unlocked.

In this subchapter, the application of ML techniques is explored to automate and enhance the analysis process of motion data, as a way of complementing the ECG analysis and thus informing when the best time is to execute noise-free ECG recordings.

The analysis of motion data using ML techniques involves several steps such as the gathering of motion data, the preprocessing of the gathered data, the extraction of features and the training of different ML models. The following subchapters describe how each one of the mentioned concerns is addressed in the context of this elderly care platform.

6.2.1 Related work

The usage of Machine Learning to analyse motion data is studied by many researchers, especially using commonly available devices like smartphones or smartwatches. Experiences with different datasets, ML models, pre-processing techniques and feature extraction modes are carried out to converge into an optimal solution concerning the recognition of different activity types and movements. This concept is often referred to as Human Activity Recognition.

An inclusive overview of HAR utilizing smartphone sensors encompasses the latest advancements in this field [73]. The survey encompasses a variety of datasets, characterized by the sampling frequency (e.g., 20 Hz, 100 Hz, 500 Hz, etc.) and the body location where the sensors are situated (e.g., waist, pants pocket, arm, etc.). Moreover, the context in which the datasets are collected can range from controlled laboratory settings to natural outdoor environments or home settings. Furthermore, some feature extraction mechanisms are proposed such as time windowing of the data, the calculation of statistical features, the calculation of the orientation-independent magnitude features and further Deep Learning methods for dimensionality reduction. In terms of ML models, the survey enumerates the Naïve Bayes, Decision Trees, Random Forests, SVMs, KNN and various neural networks.

A study from 2020 focuses on the creation of a publicly available HAR dataset from 19 subjects using smartphone sensors for several activities: driving, walking, active and inactive [74]. Also, an SVM model is put to the test for its classification capabilities on the created dataset. The accuracy and F1-score peak at 70% and 75% respectively.

Another analogous study tests the performance of the Random Forest, KNN and SVM models by classifying 6 different activities of a publicly available dataset: walking, walking upstairs, walking downstairs, sitting, standing and lying. The models achieve an accuracy of 96%, 93% and 93% respectively with 50 different features [75].

6.2.2 Dataset creation

In pursuit of creating an ML model to predict the activity type of the patient, data has to be gathered manually or from publicly available datasets. For the sake of simplicity of this proof of concept, only three different types of activity are considered: inactive, active and walking. Also, the data is manually gathered and labelled through an Android smartphone application that is built for the purpose. All recordings are performed by the author executing daily activities.

Table 5 describes each activity that is recorded:

Table 5 – Description of each recorded activity

Activity / Label	Description
Inactive	Recorded when lying quietly in bed or sitting quietly on a chair
Active	Recorded when performing everyday tasks at home (e.g., washing the dishes, cooking, going from one room to another, etc.)
Walking	Recorded when performing walking for longer periods outside of the home (e.g., walking the dog)

Each activity is recorded for 15 minutes at a frequency of 50 Hz where data from the accelerometer, gyroscope and GPS are gathered. The smartphone is held inside the pants pocket of the right leg in the portrait position pointing upwards. Each distinct activity data is saved in an independent CSV file for further analysis

Table 6 describes the data that is fetched from each sensor:

Table 6 – Description of the variables that were gathered from each sensor

Sensor type	Variable	Description	Unit of measure
Accelerometer	X	Acceleration force along the x-axis (including gravity) [76].	m/s ² (meters per second squared)
	Y	Acceleration force along the y-axis (including gravity) [76].	
	Z	Acceleration force along the z-axis (including gravity) [76].	
Gyroscope	X	Rate of rotation around the x-axis [76].	rad/s (radian per second)
	Y	Rate of rotation around the y-axis [76].	
	Z	Rate of rotation around the z-axis [76].	
GPS	Speed	The speed at a given time instance [77].	m/s (meters per second)
	Accuracy	Estimated horizontal accuracy radius in meters of this location at the 68th percentile confidence level [77].	% (percentage)

6.2.3 Pre-processing and feature extraction

The gathered motion data is not subjected to any relevant pre-processing techniques. However, the data is subjected to some feature extraction techniques like time windowing, statistical calculations and the measurement of the angular velocity of the sensors.

6.2.3.1 Time window

At a frequency of 20 Hz, a value from each sensor variable is obtained every 50 milliseconds. However, a minimum range of 3 seconds is needed to analyse the data and recognize human activities like walking [78]. Also, some studies use a fixed 2.56-second window and get accurate enough results [73], [75]. A previous survey indicates that the time window should reside between 2 and 5 seconds [73] but the best time window size is still debated in literature.

In this analysis, a fixed 5-second time window length is set to partition the motion data. For 15 minutes of activity recording, it results in 180 sliding windows of 5 seconds.

6.2.3.2 Calculated features

The accelerometer and gyroscope sensors can provide information about gravitational acceleration and angular velocity respectively. The three X, Y and Z axes values of each sensor can be reduced to a calculated magnitude value that can be obtained by the following formula:

$$\text{magnitude} = \sqrt{x^2 + y^2 + z^2}.$$

The magnitude of a signal is a feature that is extracted from multi-dimensional sensor data and is independent of orientation [73]. For each time window, the arithmetic mean of the magnitude for each sensor is calculated using the following formula:

$$\text{mean of the magnitude within a time window (5 seconds)} = \frac{\sum_{i=1}^n \sqrt{x_i^2 + y_i^2 + z_i^2}}{n}$$

n = number of sliding windows

The GPS can provide information about the geographical activity of the patient with the speed and accuracy variables. The speed informs how fast a person is moving which can facilitate the detection of prolonged walking activities. The accuracy informs how accurate and reliable the GPS signal is which can help detect if someone is inside a building or not. For each sliding window, the arithmetic mean of the speed and accuracy variables are calculated.

6.2.3.3 Feature vector

In Table 7, the four features that are used to feed the ML classifiers are listed. Each feature represents a time window of 5 seconds:

Table 7 – List of features used to feed the ML classifiers

Feature	Sensor
Acceleration force Magnitude Mean	Accelerometer
Angular velocity Magnitude Mean	Gyroscope
Speed Mean	GPS
Accuracy Mean	

6.2.4 Model training

The Random Forest classifier is employed to evaluate the predictive abilities of inactive, active, and walking activities. Its recurrent mentions in literature [73] and proven efficacy in comparable studies [75] underscore the significance of assessing the Random Forest's potential.

In pursuit of finding the best Random Forest parametrization for training, a grid search approach is conducted to discover the hyperparameter combination that is more optimized for this study. The grid search combination of hyperparameters is the following:

- a) Number of estimators: 100, 200 and 300;
- b) Maximum depth: default, 5 and 10;
- c) Minimum samples split: 2, 5, 10.

The grid search consists of a combinatory analysis of all parameters as a way of finding the most performant combination. Also, 10-fold cross-validation is applied to evaluate the performance of different data splits during the grid search analysis.

6.2.4.1 Training and testing environment

In this analysis, Python [32] is used with the Jupyter Notebook environment [33] to process the datasets, generate the feature vectors and conduct the grid search.

To create the Random Forest classifier, the “scikit-learn” library is used [70].

To the extent of homogenizing the training and testing environment for all models and feature vector combinations, the same procedures as in the ECG analysis are done to avoid biases from different training and testing instances.

Firstly, the random seed is set to an arbitrary value of 42 in Python. Also, the same value is used to pass as the random state argument for reading the datasets and executing the splitting of the training and test data. Additionally, when executing the training, the random state of 42 is also passed as an argument.

The splitting of the train and test data is set to a ratio of 70% and 30% respectively.

6.2.5 Results and discussion

The conducted grid search results in a Random Forest classifier with an accuracy of 95 %, precision, recall and F1-score of 96%. The scores can be visualized in Figure 15. This result concludes the capabilities of the Random Forest classifier in predicting the activities of active, inactive and walking. The hyperparameters that permitted the achievement of those scores are:

- a) Number of estimators: 100;
- b) Maximum depth: default;
- c) Minimum samples split: 2.

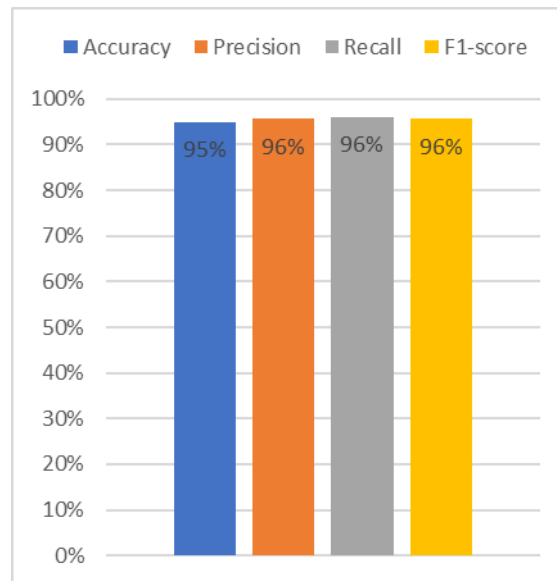


Figure 15 – Performance metrics of the Random Forest classifier

The elevated accuracy of 95 % signifies that the classifier accurately predicted nearly every instance within the test set across all activity categories. Nevertheless, it's important to note that relying solely on accuracy can be misleading, as the classifier might excel in certain labels while not performing as effectively in others.

The further interpretation of the precision and recall scores of 96 % permits us to conclude that the classifier possesses a low false positive rate as well as a low false negative rate. A low false positive rate means that when the model predicts a particular activity, it is very likely that it is accurate. On the other hand, a low false negative rate means that the model is proficient in identifying all instances of a given activity. To put it all together, the F1-score is the harmonic mean of the precision and recall provides insight into the model's ability to maintain equilibrium between correct positive classifications and minimizing false negatives and false positives.

The confusion matrix depicted in Figure 16 provides a visual representation of the performance within each specific class:

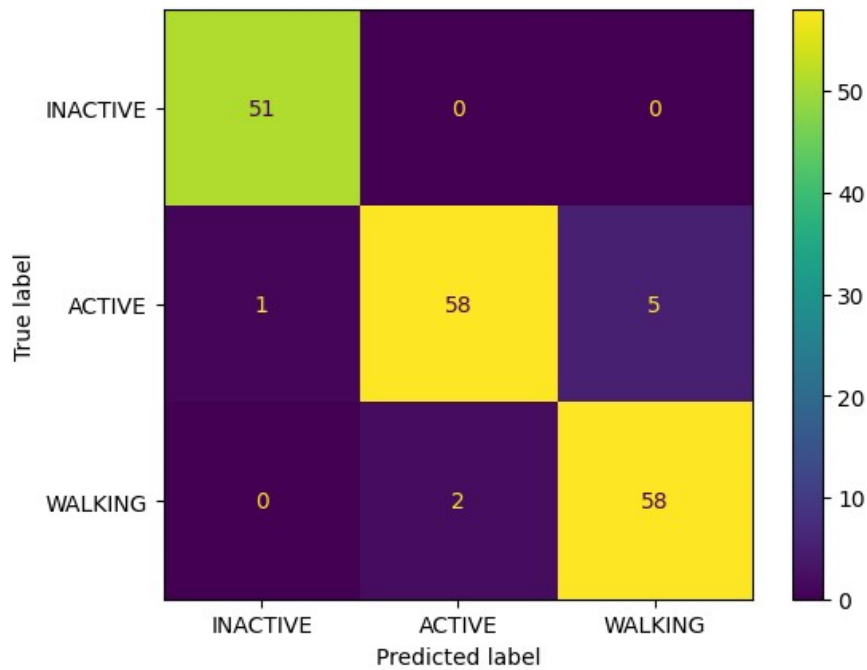


Figure 16 – Confusion matrix of the predicted labels of inactive, active and walking activities.

The analysis reveals that the "inactive" label demonstrates the most accurate predictions, as the classifier consistently performs without errors in this case. The "active" class is occasionally misclassified once as "inactive" and five times as "walking." This occurrence can be attributed to the positioning of the "active" label between the other two categories in terms of motion intensity. While it possesses more energy than "inactive," it falls short of the vigour associated with "walking." Variations in body movement amplitude during "active" instances could lead to confusion with the amplitude typical of "walking." Lastly, the "walking" activity encounters only two instances of confusion with the "active" label. This can be rationalized when considering instances where walking steps occur at a slower pace than usual, resulting in reduced speeds that might not distinctly resemble typical walking patterns.

In conclusion, employing the Random Forest as the HAR classifier proves to be a sound selection for this PoC. Nonetheless, some margin for enhancement remains in certain areas. Enhancing data collection in terms of both quantity and demographic diversity could introduce intricate patterns to the classifier, bolstering its generalizability since the data collection of only one person, in this case, might suggest potential overfitting. The validation using other devices, such as smartwatches or dedicated IoT sensors, could further help to create more variety in terms of data sources. Additionally, refining the feature extraction process offers potential for improvement, including exploring novel feature calculations and the incorporation of supplementary sensors to augment existing data.

7 IMPLEMENTATION OF THE ELDERLY CARE PLATFORM

The evolution of the presented work encompasses two discrete stages. The primary stage entails the creation of ML models dedicated to classifying ECG and motion data. Subsequently, the secondary stage involves the implementation of software modules that embody the elderly care platform, followed by the integration of the trained ML modules into the said platform. In this chapter, we will delve into the scope of the proof of concept (PoC) that will be developed as part of this internship project. It is essential to acknowledge that the envisaged platform has vast potential and encompasses numerous possibilities for future expansion and development. However, for this internship, specific limitations are set on the scope of the project to focus on the creation of a well-defined proof of concept.

7.1 Scope of the proof of concept

Given the limited duration and resources of this internship, it becomes impractical to tackle the entire expanse of the platform. Instead, the focus must be pragmatically narrowed to specific aspects, making the development of a proof of concept feasible within the given timeframe.

The PoC follows four principles, with them being feasibility, clarity, validation, and future development. These principles shape the PoC, ensuring achievability, precision, and viability. Successful implementation of the PoC will also lay the groundwork for the platform's growth and evolution.

- a) **Feasibility:** By concentrating on a specific segment of the platform, the PoC remains achievable within the allotted time and resources.
- b) **Clarity:** A well-defined scope enables to maintenance of clarity and precision throughout the development process, ensuring a clear understanding of the goals and objectives.
- c) **Validation:** The proof of concept acts as a validation mechanism to demonstrate the viability and effectiveness of the selected features or functionalities for further development of newer ones.
- d) **Future Development:** The successful implementation of this limited scope can pave the way for future expansion, guiding the platform's growth and evolution.

7.1.1 Features

The proof of concept is based on a limited set of features, but the vision is to expand with more functionalities in the future. The list of features is outlined as follows

ECG Analysis (F1): Gather data from a lead I ECG sensor and use Machine Learning to analyze ECG normality.

Motion Analysis (F2): Collect motion data via Android smartphone sensors and analyze the type of activity being performed by the patient.

Sensor Simulation (F3): Ability to simulate physical sensors for testing purposes.

Orchestrator (F4): A software module to coordinate and manage all sensors and data analysis.

Simplified GUI (F5): A user interface to visualize the implemented features for demonstration purposes.

Centralized communication over MQTT (F6): All communication between software components (sensors, Orchestrator, GUI, etc.) is managed by a centralized MQTT broker using the publish and subscribe paradigm.

7.1.2 Requirements

As a way of tracking and validating the individual functionalities, the features of the PoC are split into several individual requirements that are outlined in Table 8:

Table 8 – List of requirements

ID	Requirement	Description
F1-R1	Creation of several ML models for ECG analysis: Random Forests, Neural Networks, KNNs and EBMs	Several ML models shall be created for ECG analysis with different types of classifiers and different types of feature extraction methods for the sake of comparison purposes. The classifiers must predict whether an ECG recording is normal or abnormal.
F1-R2	Creation of a pre-processing pipeline for ECG before analysis	A fixed pipeline must be defined to pre-process the ECG data before analysis. This includes downsampling, filtering of the signal and feature extraction.
F1-R3	Integration of a physical 1-lead ECG sensor from VitalSigns [79]	The physical 1-lead ECG sensor device from the VitalSigns vendor should be integrated into the system to be used for demonstration purposes.
F1-R4	The ECG data should be recorded using time windows of 10 seconds	The ECG sensor should record the data for 10 seconds and should only send the data for analysis if the 10-second recording is performed with success.

F1-R5	The ECG data should be sent for analysis at a frequency of 250 Hz	Independently of the physical device's default frequency, the data should be sent over the network at a 250 Hz frequency for further analysis as a way of saving bandwidth usage.
F2-R1	Creation of an ML model for motion analysis using Random Forests	An ML model must be created to receive motion data as input and classify the activity that is being performed by the patient. The activities are: inactive, active and walking.
F2-R2	Creation of a pre-processing pipeline for motion data before analysis	A fixed pipeline must be defined to extract features from motion data before analysis.
F2-R3	Creation of an Android application to gather data from the accelerometer, gyroscope and GPS sensors	An Android application is needed to gather motion data for the sake of this proof of concept. The application shall gather data about the 3 axes of the gyroscope and accelerometer, and the speed and accuracy provided by the GPS sensor.
F2-R4	Recording of motion data at 20 Hz	The frequency of the motion recording shall be 20 Hz only due to bandwidth optimization and device energy savings.
F2-R5	The time window of motion data should be 5 seconds	Motion data shall be gathered for 5 seconds and then sent for analysis if the 5-second recording is successful.
F3-R1	Simulation of ECG data	It shall be possible to simulate the sending of ECG data for testing purposes. The data must represent realistic or pre-gathered ECG recordings. In other words, it should be possible to test the ECG-related modules without a physical ECG sensor.
F3-R2	Simulation of motion data	It shall be possible to simulate the sending of motion data for testing purposes. The data must represent realistic or pre-gathered motion recordings. In other words, it should be possible to test the motion-related modules without a physical motion sensor or smartphone.

F4-R1	The analysis of all sensor data shall be performed by an independent software module, named Orchestrator	To the extent of separation of concerns, the Orchestrator must receive the data from all sensors (ECG, motion, etc.) and execute the tasks of pre-processing the data, extracting features and feeding to the ML models.
F3-R3	Inside the Orchestrator, the analysis of sensor data must be divided into individual and independent software modules named “Assessors”, e.g. ECG Assessor, Motion Assessor, etc.	An Assessor component consists of a piece of software that analyses a certain type of data (ECG, motion, etc.). For example, an Assessor for ECG analysis only takes care of incoming ECG data and its respective analysis, discarding therefore any other data type from other sensors. This concept of “Assessor” is important for the separation of concerns and guarantees the expandability of the system for incoming new types of sensors.
F4-R2	The orchestrator shall publish the outcome of the ML classifiers (ECG and motion)	All predictions from the classifiers or results from any kind of analysis shall be published over MQTT as a way of being visible to other software modules.
F5-R1	The GUI should have information about the sensors in the form of charts for motion data and ECG data	The charts should plot the values that are retrieved from the sensors or their calculated features when applicable. Each sensor or data type should have its distinct separate chart.
F5-R2	It should be possible to change the classifier that is being tested for the ECG analysis	The GUI should have graphical elements that permit the selection of the different kinds of ML models and feature extraction modes for the ECG data analysis.
F5-R3	The GUI shall display the output of the motion and ECG data analysis.	The predictions of the ML classifiers and results of any other analysis should be displayed in the GUI. It can consist of simple labels that update according to the predictions made by the classifiers.

F6-R1	All recorded sensor data (ECG, motion, etc.) must be published over MQTT	The sensor data must be published over MQTT so that it can be analysed by the Orchestrator and used by the GUI for visualization.
MISC-R1	Every software module must be parameterizable through configuration files for easy manipulation	All software modules must be parametrizable with configuration files. For example, the Orchestrator must be configurable in terms of MQTT host and port, which assessors to load, MQTT topics to subscribe to, etc. Analogously, for the sensors, it should be possible to define the MQTT host, port and topics, the recording frequency, time window size, etc.

7.2 Materials and Methods

7.2.1 Python

Python [68] is chosen as the core programming language to develop this PoC due to its high compatibility with IoT, Machine Learning, APIs, libraries, and more. Its versatility allows seamless integration of IoT devices, real-time monitoring, and data collection. The robust support for Machine Learning enables easy integration of intelligent algorithms for personalized care. Python's extensive APIs facilitate efficient communication with external services, enhancing the platform's capabilities for elderly care.

The platform's core, including the Orchestrator and its Assessor subcomponents, relies entirely on Python. Additionally, it serves as the primary tool for developing sensor software modules that encapsulate physical sensors and their low-level communications over BLE such as the ECG sensor.

7.2.2 Vue.js

Vue.js is selected for the GUI of the presented PoC due to its adaptability, seamless integration with various services and APIs such as MQTT, and rich plugin ecosystem. Its support for server-side rendering and progressive web application development ensures optimal performance. With Vue.js, the platform can efficiently connect with external systems, enhancing functionalities for elderly care.

The future development of the platform is already planned to incorporate Vue.js as a crucial element, leading towards a more final and complete solution.

7.2.3 Android Studio

To overcome the absence of motion devices with accelerometers, gyroscopes, and GPS sensors, an Android Studio application is developed. This temporary solution enables data collection from a standard smartphone, providing essential motion data for the implementation of this PoC. However, the inclusion of the Android application for motion data (or other types of data) in the system's future is still uncertain.

7.2.4 MQTT

MQTT [31] is chosen for the platform due to its lightweight and efficient nature, making it ideal for seamless communication between components. Its publish-subscribe pattern ensures real-time data exchange from various sensors, such as ECG and motion sensors. With support for Quality of Service levels, MQTT ensures reliable and scalable message delivery, crucial for maintaining data integrity.

Its low bandwidth and resource requirements align well with the platform's IoT nature and focus on efficiency and expandability to new incoming types of sensors and functionalities.

7.2.4.1 EMQ X

EMQ X is a highly versatile and scalable open-source implementation of an MQTT broker that facilitates efficient communication and data exchange between IoT devices. In this project, it is used as the MQTT broker of the platform.

7.2.5 Software design patterns

By incorporating design patterns into this elderly care PoC, a robust foundation is established for future development, especially when integrating new sensors and functionalities. The implementation of design patterns enhances code maintainability and facilitates seamless expansion, ensuring a solid base for the project's advancement. The current PoC encompasses various software components, such as the Orchestrator, Assessors, and sensor drivers, exemplifying the utilization of design patterns to ensure horizontal expandability and smooth integration of additional features and sensors. All patterns used in this project are inspired by the “Dive Into Design Patterns” book [80].

Factory method: This design pattern is ideal for the Orchestrator's core in the elderly care platform as it separates the concern of creating Assessor objects. This allows for easy integration of newer Assessor types in the future, while the Orchestrator can focus solely on starting and stopping Assessors without the need to create each one individually. This enhances flexibility and maintainability in the system. Also, it permits the handling of diverse ML models for the ECGAssessor without the burden of loading models from various technologies and parameters. This enhances flexibility and modularity, allowing seamless adaptation to different

ML models for ECG assessment without impacting the ECGAssessor's responsibilities.

Adapter: The adapter pattern is essential for encapsulating MQTT communication and low-level sensor interactions in this PoC. It allows the core to remain independent of specific hardware or technologies, facilitating easy changes without disrupting the system. This promotes flexibility, adaptability, and maintainability, making the platform more resilient to future advancements and updates.

Singleton: The singleton pattern is beneficial for MQTT, logging, and other objects in the platform that must exist only once in a certain module like the Orchestrator or sensor drivers. It ensures data integrity, resource efficiency, and centralized access, preventing conflicts and promoting a coherent and efficient system.

Template method: This pattern is vital for the elderly care platform as it defines a fixed algorithm structure with customizable steps by subclasses. This promotes code reusability, maintainability, and adaptability, ensuring a consistent yet flexible approach to algorithm execution. It is very useful to implement the simulation of sensor data and systematic and agnostic message parsing and treatment in MQTT.

On top of the usage of the mentioned design patterns, the fundamental rules and principles of object-oriented programming were applied in all parts of the software.

7.3 General System Architecture

This subchapter presents the system architecture that underpins the foreseen proof of concept. The focal point of this architecture is an MQTT broker, which assumes a central role in orchestrating communication between IoT sensors and devices and the Orchestrator module. The ensuing sections expound on the intricacies of the architecture, delineate the functions of distinct components, and elucidate the orchestrator's role in data analysis and sensor control, facilitated by the Assessors and Controllers modules. In Figure 17, the general architecture diagram is illustrated.

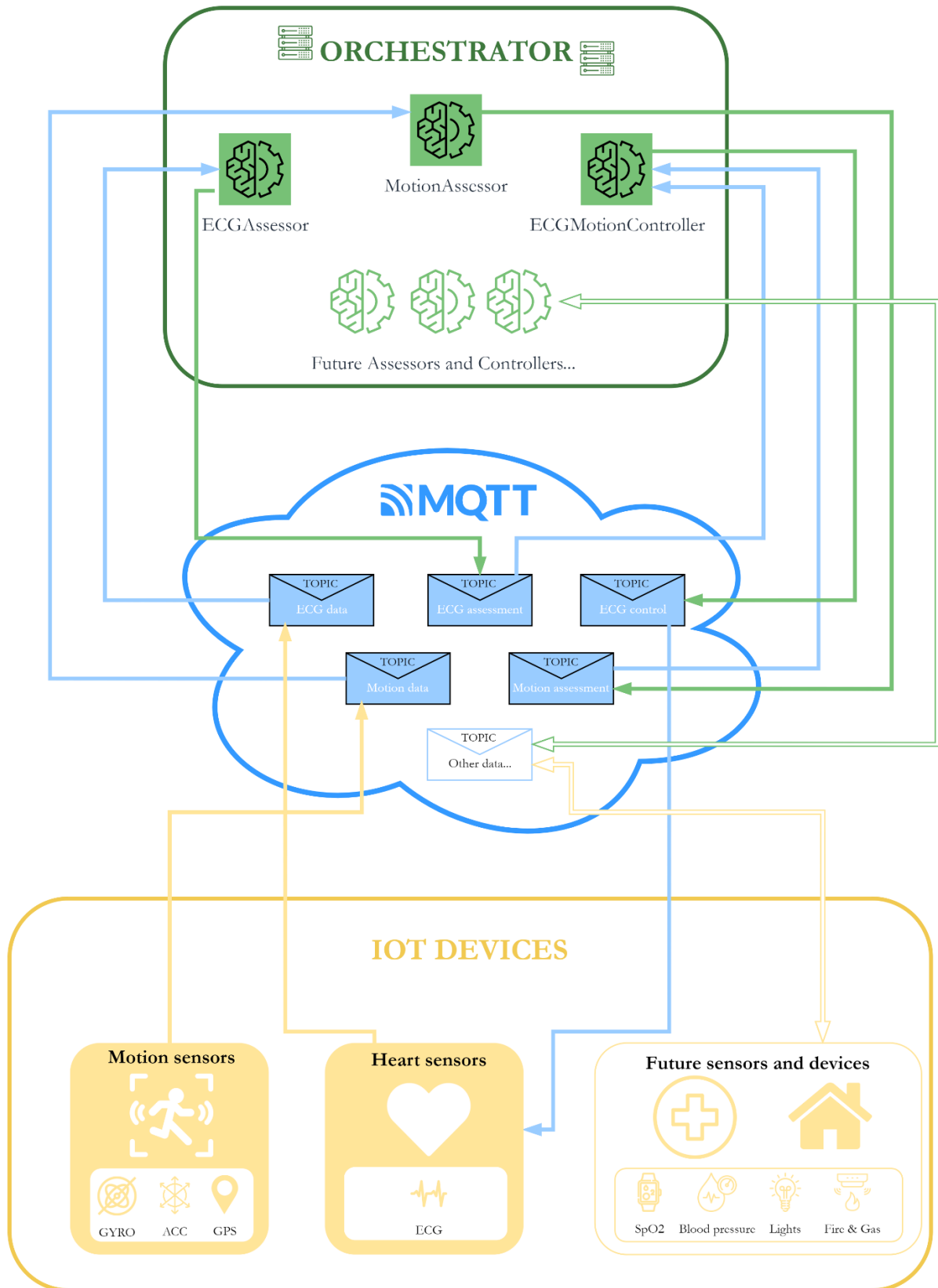


Figure 17 – General architecture diagram. Produced with draw.io software [32] and Flaticons [33].

7.3.1 Central MQTT Broker

The MQTT broker serves as a pivotal communication bridge between modules in the PoC. To facilitate seamless data exchange, five distinct topics have been established, each designated for specific purposes.

ECG Data: This topic is dedicated to transmitting raw ECG data collected from the ECG sensor. It enables the transfer of unprocessed ECG readings for further analysis.

ECG Assessment: The ECG Assessment topic is employed to disseminate the outcomes of the ECG data analysis performed by the ECGAssessor module. This entails the results generated by the ML models which determine whether the ECG readings indicate a normal or abnormal cardiac pattern.

Motion Data: Within the system, the Motion Data topic is utilized for transmitting raw motion data sourced from the accelerometer, gyroscope, and GPS sensors.

Motion Assessment: The Motion Assessment topic serves as the medium for publishing the evaluations of motion data conducted by the MotionAssessor module which indicates the patient's physical activity status, classifying them as walking, active, or inactive.

ECG Control: Finally, the ECG Control topic plays a critical role in communicating the control state of the ECG sensor. This involves determining whether the ECG sensor should be actively transmitting data or ceasing its data transmission temporarily because of motion detection.

In this manner, the MQTT broker effectively orchestrates communication through these well-defined topics, ensuring the efficient functioning of the PoC and enabling data analysis, assessment, and control of pertinent sensor information. Furthermore, the system is easily expandable to newer topics to accept data from new sensors that may integrate the elderly care platform in the future.

7.3.2 The Orchestrator module

The Orchestrator module serves as the pivotal component responsible for overseeing and managing the entire platform in terms of data collection and analysis. As the central cognitive entity, it analyses data from sensors, makes discerning decisions, and exercises control over various system components. In this PoC, two Assessors and one Controller are implemented to analyse ECG and motion data. Indeed, the Orchestrator remains open to the possibility of incorporating additional Assessors and Controllers in the future, thereby accommodating the integration of new sensors and devices as they become available.

7.3.2.1 Assessors

Prominent elements within the Orchestrator module are the specialised components known as "Assessors." Each Assessor fulfils a critical function by evaluating specific

facets of the data and producing salient metrics or assessments. Tailored to focus on distinct domains, such as environmental monitoring, performance evaluation, or anomaly detection, these Assessors contribute decisively to the comprehensive system analysis. More specifically, in this PoC, the focus lies on the ECGAssessor and MotionAssessor.

ECGAssessor: This Assessor is responsible for gathering ECG data from the respective sensor over MQTT and subjecting it to ML techniques, aiming to predict whether the recorded data exhibits a normal or abnormal pattern. Subsequently, it disseminates the assessment to other modules within the system.

MotionAssessor: This Assessor is responsible for gathering motion data from the accelerometer, gyroscope and GPS sensors over MQTT and subjecting it to ML techniques, aiming to predict the kind of activity exerted by the patient (inactive, active and walking). Subsequently, it disseminates the assessment to other modules within the system.

7.3.2.2 Controllers

Moreover, the Orchestrator encompasses "Controllers" that play a central role in effectuating decisions and controlling actuators or IoT devices based on the insights furnished by the Assessors. Malleable to dynamic circumstances, these Controllers enact corrective measures, trigger actions, and optimise system performance. In the context of this PoC, only a specific Controller is developed to control the ECG sensor device.

ECGMotionController: This Controller has the objective of managing the recording state of the ECG sensor. It uses the information provided by the MotionAssessor to detect the activity type of the patient and activates or deactivates the ECG sensor accordingly. The goal is to record ECG only when the patient is at rest so that only noise-free recordings are analysed.

7.3.3 IoT sensors and devices

This layer hosts an array of IoT sensors and devices with the primary function of collecting and transmitting diverse data, including heart and motion-related information. For this PoC, the system employs specific sensors, namely a heart ECG sensor and three motion sensors. Nevertheless, the system retains adaptability for future integrations.

Heart ECG sensor: This sensor captures time series data concerning the heart's electrical activity. The acquired data is propagated via MQTT to the ECG data topic, where it undergoes in-depth analysis by the ECGAssessor module. Additionally, the ECG sensor responds to commands from the ECGMotionController, enabling data recording activation or deactivation to conserve energy and facilitate noise-free recordings based on the physical activity of the patient.

Motion sensors: Represented by the accelerometer, gyroscope, and GPS, these sensors are instrumental in recording the patient's physical status, discerning whether they are at rest or in motion. Analogous to the ECG sensor, the motion data is transmitted via MQTT to the designated motion data topic to be further analysed by the MotionAssessor.

7.4 ECG recording and analysis workflow

The ECG recording and analysis workflow initiates within the heart sensor module. It commences by continuously capturing ECG data until a complete 10-second time window is filled. This data is subsequently disseminated over MQTT through the designated ECG data topic. The transmitted message is automatically directed to the ECGAssessor housed within the Orchestrator, as it is subscribed to the corresponding topic. Following this, the ECGAssessor engages in a series of pre-processing techniques outlined in the ECG analysis chapter. Once the data has been suitably pre-processed, it undergoes analysis by the ML models. The results yielded by these models are then published to the ECG assessment topic, rendering them accessible to other software modules for further utilization.

Figure 18 portrays the workflow of ECG recording and analysis, elucidating the step-by-step process through a diagram.

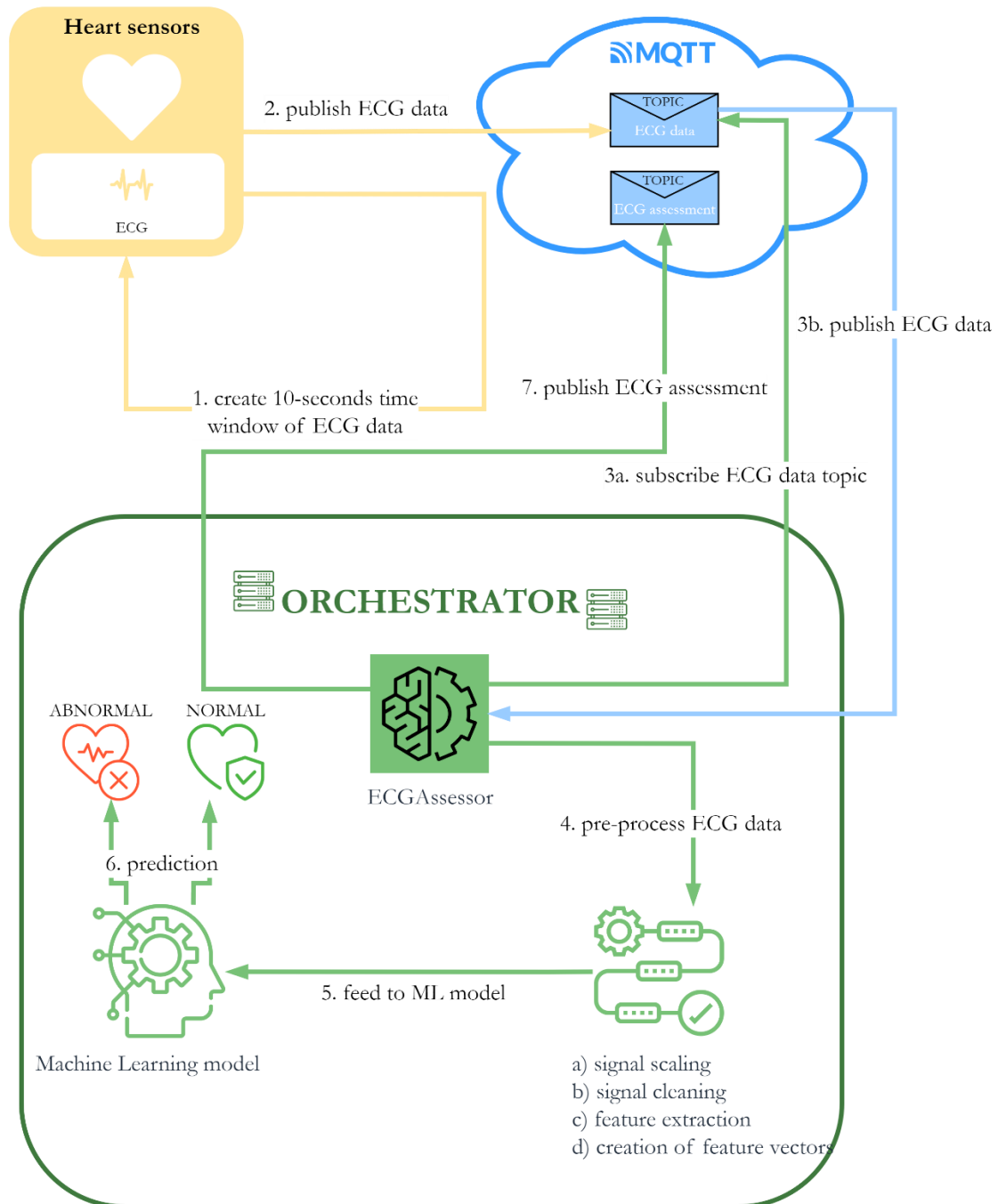


Figure 18 – ECG recording and analysis workflow diagram. Produced with draw.io software [32] and Flaticons [33].

7.5 Motion recording and analysis workflow

The motion recording and analysis workflow originates within the motion sensor module, encompassing the accelerometer, gyroscope, and GPS components. This process commences by continuously capturing motion data until a comprehensive 5-second time frame is attained. Subsequently, this data is disseminated via MQTT through the designated motion data topic. The transmitted message seamlessly

reaches the MotionAssessor component residing within the Orchestrator, facilitated by its subscription to the corresponding topic. Following this initial step, the MotionAssessor undertakes a sequence of pre-processing techniques outlined in the Motion analysis chapter. Once the data has undergone suitable pre-processing, it undergoes analysis by dedicated ML models. The outcomes derived from these models are subsequently published to the motion assessment topic, effectively making them accessible to other software modules for further application and utilization.

Figure 19 portrays the workflow of motion recording and analysis, elucidating the step-by-step process through a diagram.

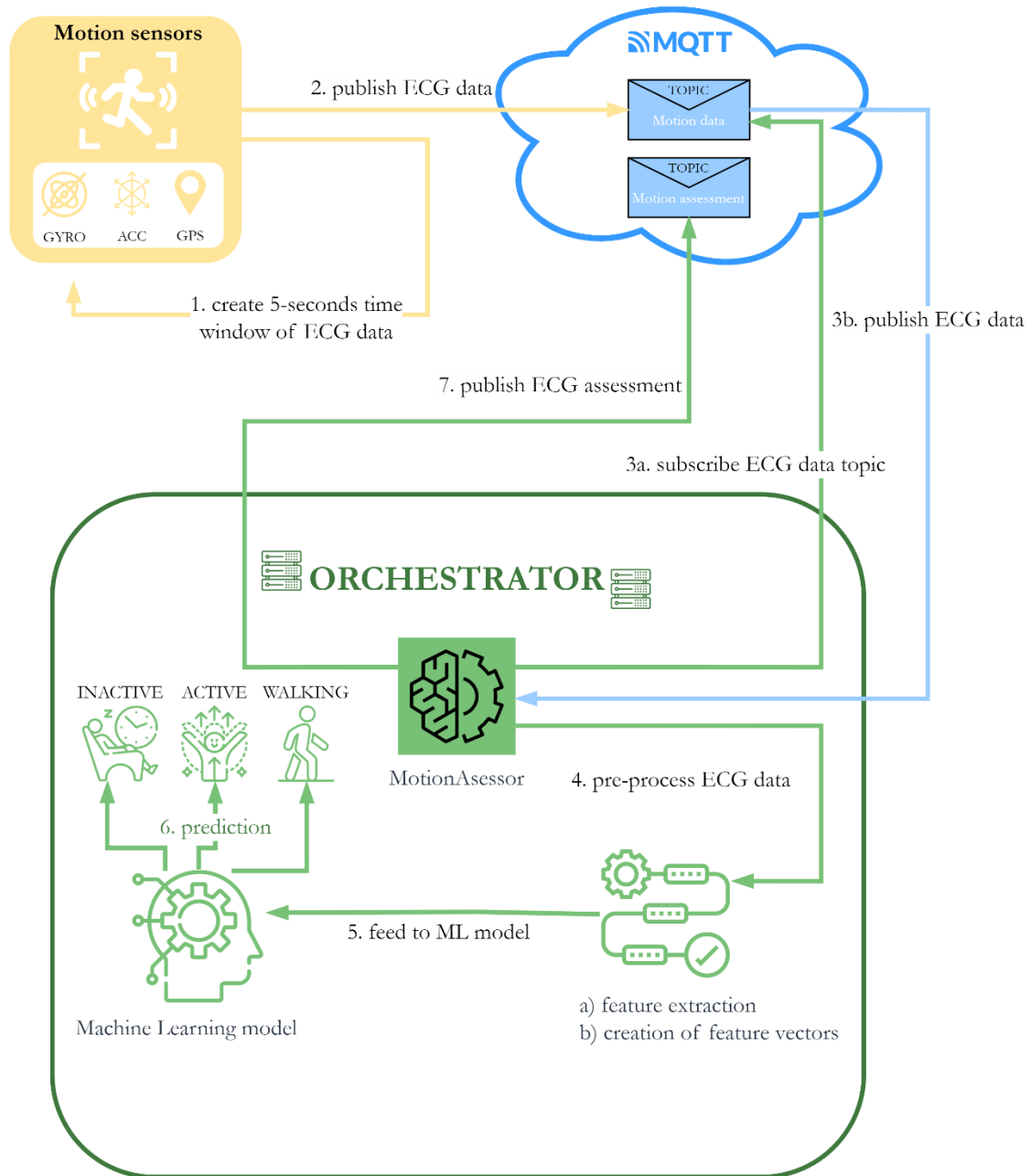


Figure 19 – Motion recording and analysis workflow diagram. Produced with draw.io software [32] and Flaticons [33].

7.6 ECG control workflow

The ECG control workflow is initiated by capturing motion data through the motion sensors. Upon collecting a 5-second duration of motion data, it is transmitted via MQTT to the designated motion data topic. The MotionAssessor, as elaborated in the preceding subchapter, evaluates this motion data and subsequently publishes its assessment into the respective MQTT topic. The ECGMotionController component, subscribed to the motion assessment topic, responds dynamically to

each new motion assessment. When an assessment indicates inactivity, the ECGMotionController dispatches an "on" message to the ECG control topic. Conversely, in the case of activity, an "off" message is dispatched. Ultimately, this information is harnessed by the ECG sensor to govern the activation or deactivation of recording accordingly.

Figure 20 portrays the workflow of ECG control based on the patient's motion, elucidating the step-by-step process through a diagram.

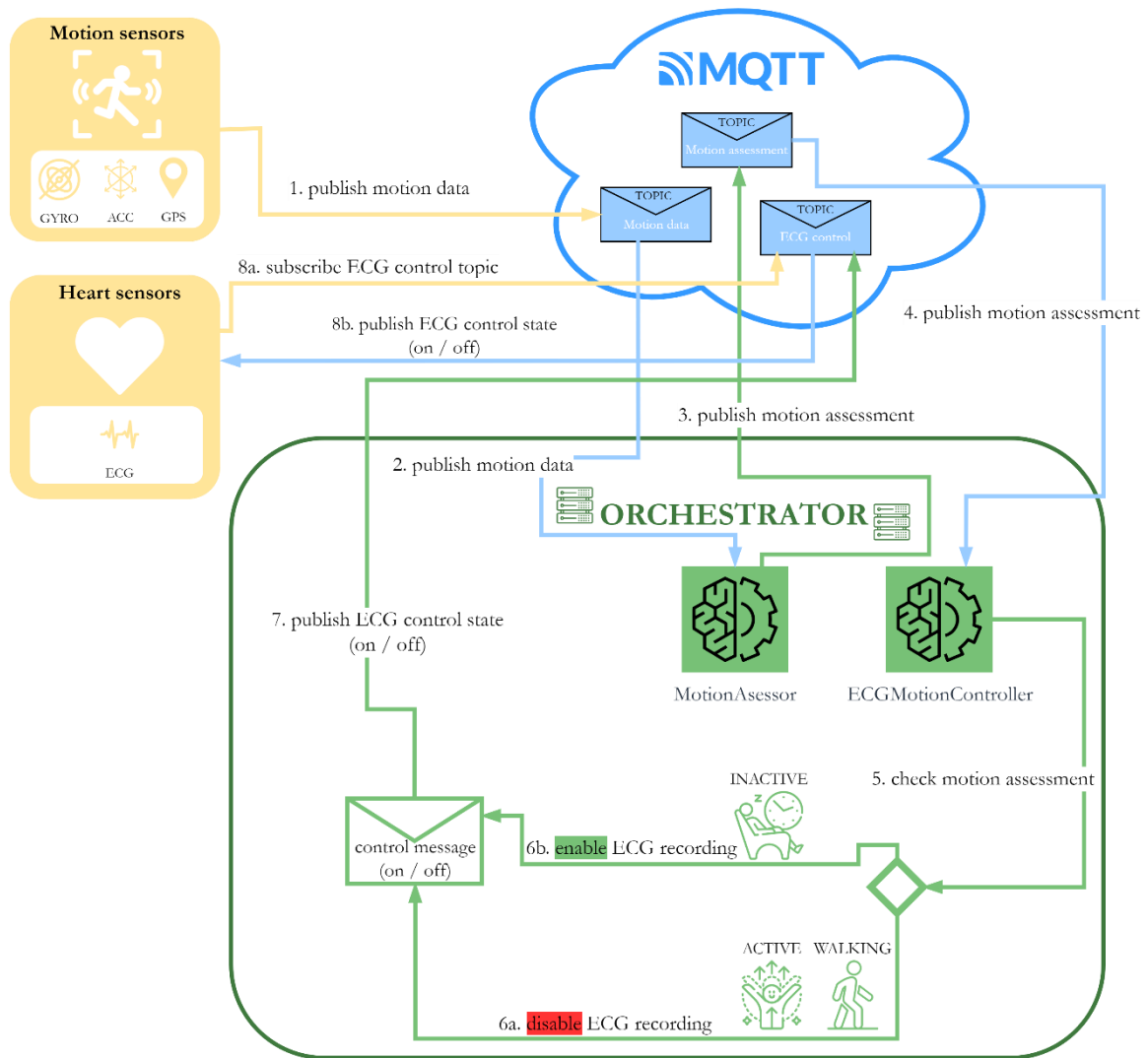


Figure 20 – ECG control workflow diagram. Produced with draw.io software [32] and Flaticons [33].

7.7 Dashboard

The development of a dashboard with graphical capabilities for monitoring patients is of paramount importance, given that the primary objective of this elderly care solution is to assist and monitor the elderly. The dashboard's key functionality should

encompass the visualization of all interconnected sensors and their corresponding analysis provided by the Orchestrator components.

In this PoC, a simplified dashboard has been defined to facilitate interaction with the currently developed components. Nevertheless, its architecture has already been designed to seamlessly accommodate the integration of new data types in the future. Access to the system's information is facilitated through MQTT, following a similar approach as with the other modules.

Within this simplified dashboard, two distinct view components have been implemented, each with a specific focus on presenting particular types of data. One component is dedicated to displaying ECG data along with their corresponding analysis, while the other component showcases motion information alongside their respective analysis. The contents inside each component vary from charts, text labels and any kind of visual component that helps to visualize the data. Figure 21 showcases how the GUI interacts with the system through MQTT:

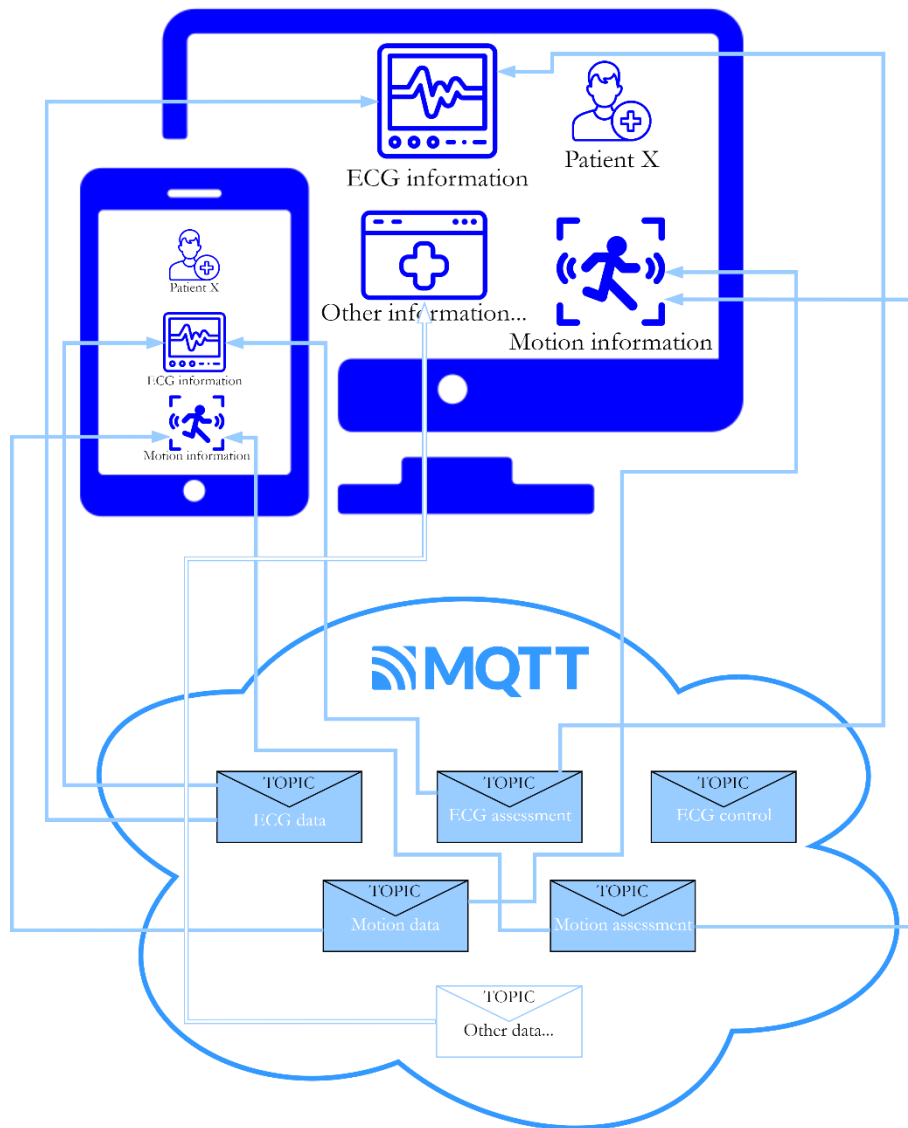


Figure 21 – Dashboard and MQTT communication diagram. Produced with draw.io software [32] and Flaticons [33].

Figure 22 depicts how the GUI shows the information about ECG and motion data that is being recorded and their respective assessments.

IoT Platform for Personal Healthcare of Elderly People



Figure 22 – Screenshot of the simple GUI to visualize ECG and motion data.

8 CONCLUSIONS AND FUTURE WORK

In this project, the creation of an IoT platform tailored to address the healthcare needs of elderly individuals is studied. Due to the vastness of functionalities and features that are possible to integrate into the overall system, a scope-closed proof of concept is established. By focusing on the development of a proof of concept, it is possible to validate the applicability and viability of the system through the implementation and testing of a fixed subset of functionalities. This approach allows us to lay a foundational framework for a comprehensive healthcare solution that integrates IoT technology, advanced data analysis, and real-time monitoring.

The ECG analysis employing ML techniques demonstrates the feasibility of constructing classifiers for evaluating ECG recording normality, aligning with prior research in literature. Nevertheless, substantial room for enhancement persists. Mainly, despite the vastness of the used training datasets, their quality and variety could be improved. By partnering with a cardiologist or trying to find new sources of ECG data, it would be possible to expedite outlier identification and elimination or even assess the truth of the labels, resulting in a more refined dataset mirroring real-world conditions. Moreover, to extract more nuanced insights from the ECG data, a deeper exploration of feature extraction methods could be beneficial. Despite achieving satisfactory results in this study, the performance is still not enough to be used in medical contexts. Moreover, hyperparameter tuning permits the performance enhancement of both trained neural networks and traditional classifiers. Unfortunately, the constrained computational resources in this study precluded the execution of hyperparameter tuning techniques like grid search and therefore didn't permit the achievement of the most optimal results possible. Nonetheless, considering the satisfactory performance of the trained classifiers, the potential for enhancement is evident. Systematically exploring diverse hyperparameters holds the promise of refining the models' capacity to precisely discern among different ECG recordings, thus elevating the accuracy of our evaluations.

The motion analysis undertaken in this study has attained promising results with an accuracy of 95% through the implementation of the Random Forest classifier. This outcome underscores the classifier's potential to effectively predict several motion-based activities such as inactive, active, and walking. However, a closer look reveals areas for further development and refinement. The dataset employed in this analysis encompasses 15-minute activity recordings authored only by the author of the study and proves to be enough for optimal results. Nevertheless, the dataset's limited scope presents expansion opportunities. To enhance the generalizability and robustness of the classifier, an effort to gather data from diverse sources, devices, and user demographics is imperative. Also, the integration of datasets available in the literature could be beneficial if undergone through a process of homogenisation to be compatible with the motion data structure in this project. By incorporating

data from various devices and capturing a broader demographic spectrum, the classifier's ability to accurately predict activities across different scenarios can be significantly improved. Furthermore, the variety of activity prediction can be further expanded. While the current model adeptly differentiates between inactive, active, and walking states, its scope can be broadened by introducing a more comprehensive array of activities such as fall detections, among others. This expansion could further help assess the patient's activity intensity throughout the day and help detect more situations of danger like falls.

8.1 Future work

While this study successfully demonstrated the feasibility of employing ML techniques for ECG analysis, further improvements are possible to enable more comprehensive cardiac health assessment. Instead of merely classifying ECG recordings as normal or abnormal, future work could focus on developing classifiers capable of identifying specific types of heart conditions. Collaborating with cardiologists and acquiring high-quality and diverse ECG datasets would allow for refining the classifier's accuracy and reliability. Additionally, the integration of an intelligent pre-classification module directly within the ECG sensor unit is a promising direction. By implementing a preliminary classification step on the sensor itself, only abnormal recordings would be transmitted to the central orchestrator for in-depth analysis. This approach minimizes data transmission overhead and optimizes the allocation of computational resources. Expanding the ECG sensor capabilities by incorporating multi-lead ECG recordings is crucial for more accurate cardiac evaluations. The integration of advanced ECG sensors with multiple leads will enable a comprehensive assessment of cardiac activity, enhancing the system's ability to detect and classify various heart anomalies. To further enhance the motion analysis capabilities of the system, exploring more complex activities, particularly fall detection, presents a valuable direction for future work. Developing and integrating algorithms that can accurately identify fall events based on motion data would greatly improve patient safety and emergency response.

Moreover, transitioning from the current Android application to dedicated IoT sensors optimized for motion analysis is recommended. These specialized sensors would provide higher accuracy, real-time monitoring, and better compatibility with the broader IoT framework. This change would facilitate continuous and unobtrusive monitoring of elderly individuals' activities, contributing to their overall well-being.

Lastly, the modular nature of the developed software and feature extraction methods allows for seamless integration of additional health sensors such as SpO₂, blood pressure, and respiratory rate monitors. By incorporating a wider range of health data, the system can provide a more holistic view of an individual's well-being, enabling more comprehensive health assessments and interventions. Furthermore,

extending the system's capabilities beyond healthcare to encompass domotic devices and sensors is a natural progression. Integrating sensors for environmental monitoring, presence detection, gas and fire detection, and lighting control would empower the system to not only ensure medical safety but also enhance the overall living conditions and security of elderly individuals.

In conclusion, the proof of concept presented in this study lays a solid foundation for the development of a comprehensive IoT-based healthcare solution tailored to the needs of elderly individuals. The future work outlined here highlights the potential for refining and expanding the system's capabilities, enhancing its accuracy, reliability, and usability. By further developing the ECG and motion analysis components, incorporating advanced sensors, and expanding into the realm of domotics, the proposed system has the potential to revolutionize elderly care, promoting both health and quality of life.

REFERENCES

- [1] ‘Ageing’. Accessed: Aug. 08, 2023. [Online]. Available: https://www.who.int/health-topics/ageing#tab=tab_1
- [2] D. J. Aekanth Suhas G. and Tillinghast, ‘The Emergence of Wearable Technologies in Healthcare: A Systematic Review’, in *Human-Automation Interaction: Mobile Computing*, M. and R. P.-L. P. and T. M. M. Duffy Vincent G. and Ziefle, Ed., Cham: Springer International Publishing, 2023, pp. 43–59. doi: 10.1007/978-3-031-10788-7_3.
- [3] ‘ISEC - Instituto Superior de Engenharia de Coimbra | Instituto Politécnico de Coimbra’. Accessed: Aug. 31, 2023. [Online]. Available: <https://www.isec.pt/PT/Default.aspx>
- [4] ‘Kirepo’. Accessed: Aug. 31, 2023. [Online]. Available: <https://kirepo.lu/>
- [5] S. B. Baker, W. Xiang, and I. Atkinson, ‘Internet of Things for Smart Healthcare: Technologies, Challenges, and Opportunities’, *IEEE Access*, vol. 5, pp. 26521–26544, Nov. 2017, doi: 10.1109/ACCESS.2017.2775180.
- [6] J. I. Abe, I. O. Igagwu, and others, ‘Internet of things (IOTs) Based-Domotic Engineering’, 2023, Accessed: Aug. 10, 2023. [Online]. Available: https://www.academia.edu/download/98260557/Internet_of_things_IOTs_Based_Domotic_Engineering.pdf
- [7] ‘What are Message Brokers? | IBM’. Accessed: Aug. 11, 2023. [Online]. Available: <https://www.ibm.com/topics/message-brokers>
- [8] A. Lazidis, K. Tsakos, and E. G. M. Petrakis, ‘Publish–Subscribe approaches for the IoT and the cloud: Functional and performance evaluation of open-source systems’, *Internet of Things*, vol. 19, p. 100538, 2022, doi: <https://doi.org/10.1016/j.iot.2022.100538>.
- [9] ‘A Brief History of AI’. Accessed: Oct. 05, 2023. [Online]. Available: <https://aitopics.org/misc/brief-history>
- [10] S. E. Haupt *et al.*, ‘The History and Practice of AI in the Environmental Sciences’, *Bull Am Meteorol Soc*, vol. 103, no. 5, pp. E1351–E1370, 2022, doi: <https://doi.org/10.1175/BAMS-D-20-0234.1>.
- [11] A. L. Samuel, ‘Some Studies in Machine Learning Using the Game of Checkers’, *IBM J Res Dev*, vol. 3, no. 3, pp. 210–229, Jul. 1959, doi: 10.1147/rd.33.0210.
- [12] J. Bell, ‘What is machine learning?’, *Machine Learning and the City: Applications in Architecture and Urban Design*, pp. 209–216, May 2022, doi: 10.1007/978-3-319-18305-3_1/COVER.

- [13] ‘CP-04 - Artificial Intelligence Tools and Platforms for GIS | GIS&T Body of Knowledge’. Accessed: Oct. 06, 2023. [Online]. Available: <https://gistbok.ucgis.org/bok-topics/2023-quarter-01/artificial-intelligence-tools-and-platforms-gis>
- [14] ‘Machine Learning model training over time’. Accessed: Oct. 21, 2023. [Online]. Available: <https://www.langton.cloud/machine-learning-model-training-over-time/>
- [15] R. Caruana and A. Niculescu-Mizil, ‘An Empirical Comparison of Supervised Learning Algorithms’, in *Proceedings of the 23rd International Conference on Machine Learning*, in ICML ’06. New York, NY, USA: Association for Computing Machinery, 2006, pp. 161–168. doi: 10.1145/1143844.1143865.
- [16] L. Breiman, ‘Random Forests’, *Mach Learn*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
- [17] P. Cunningham and S. J. Delany, ‘k-Nearest Neighbour Classifiers: 2nd Edition (with Python examples)’, *ACM Comput Surv*, vol. 54, no. 6, Apr. 2020, doi: 10.1145/3459665.
- [18] H. Nori, S. Jenkins, P. Koch, and R. Caruana, ‘InterpretML: A Unified Framework for Machine Learning Interpretability’, Sep. 2019, [Online]. Available: <http://arxiv.org/abs/1909.09223>
- [19] D. Jakhar and I. Kaur, ‘Artificial intelligence, machine learning and deep learning: definitions and differences’, *Clin Exp Dermatol*, vol. 45, no. 1, pp. 131–132, Jan. 2020, doi: 10.1111/CED.14029.
- [20] ‘Artificial Neural Networks - The Neuron - Blogs - SuperDataScience | Machine Learning | AI | Data Science Career | Analytics | Success’. Accessed: Oct. 19, 2023. [Online]. Available: <https://www.superdatascience.com/blogs/artificial-neural-networks-the-neuron>
- [21] A. K. Jain, J. Mao, and K. M. Mohiuddin, ‘Artificial neural networks: a tutorial’, *Computer (Long Beach Calif)*, vol. 29, no. 3, pp. 31–44, Mar. 1996, doi: 10.1109/2.485891.
- [22] Wikipedia contributors, ‘Deep learning — Wikipedia, The Free Encyclopedia’. 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Deep_learning&oldid=1173423104
- [23] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. ‘O’Reilly Media, Inc.’, 2022.
- [24] Wikipedia contributors, ‘Residual neural network — Wikipedia, The Free Encyclopedia’. 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Residual_neural_network&oldid=1164824154

- [25] S. V. Stehman, ‘Selecting and interpreting measures of thematic classification accuracy’, *Remote Sens Environ*, vol. 62, no. 1, pp. 77–89, Oct. 1997, doi: 10.1016/S0034-4257(97)00083-7.
- [26] D. M. W. Powers, ‘Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation’, 2008. [Online]. Available: <https://api.semanticscholar.org/CorpusID:55767944>
- [27] ‘Confusion Matrix Generator’. Accessed: Oct. 21, 2023. [Online]. Available: <https://www.damianoperri.it/public/confusionMatrix/>
- [28] Wikipedia contributors, ‘Accuracy and precision — Wikipedia, The Free Encyclopedia’. 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Accuracy_and_precision&oldid=1170544823
- [29] Wikipedia contributors, ‘Precision and recall — Wikipedia, The Free Encyclopedia’. 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Precision_and_recall&oldid=1163655774
- [30] Wikipedia contributors, ‘F-score — Wikipedia, The Free Encyclopedia’. 2023. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=F-score&oldid=1171586319>
- [31] ‘MQTT Version 5.0’. Accessed: Jul. 26, 2023. [Online]. Available: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>
- [32] ‘draw.io’. Accessed: Aug. 01, 2023. [Online]. Available: <https://www.drawio.com/>
- [33] ‘Vector Icons and Stickers - PNG, SVG, EPS, PSD and CSS’. Accessed: Aug. 01, 2023. [Online]. Available: <https://www.flaticon.com/>
- [34] A. J. Perez, F. Siddiqui, S. Zeadally, and D. Lane, ‘A review of IoT systems to enable independence for the elderly and disabled individuals’, *Internet of Things*, vol. 21, p. 100653, Apr. 2023, doi: 10.1016/J.IOT.2022.100653.
- [35] S. Y. Y. Tun, S. Madanian, and F. Mirza, ‘Internet of things (IoT) applications for elderly care: a reflective review’, *Aging Clin Exp Res*, vol. 33, no. 4, pp. 855–867, Apr. 2021, doi: 10.1007/S40520-020-01545-9/TABLES/4.
- [36] R. Das, A. Tuna, S. Demirel, and M. K. Yurdakul, ‘A Survey on the Internet of Things Solutions for the Elderly and Disabled: Applications, Prospects, and Challenges’, *International Journal of Computer Networks And Applications*, vol. 4, no. 3, p. 1, Jun. 2017, doi: 10.22247/ijcna/2017/49023.
- [37] N. M. López, S. Ponce, D. Piccinini, E. Perez, and M. Roberti, ‘From Hospital to Home Care: Creating a Domotic Environment for Elderly and

Disabled People’, *IEEE Pulse*, vol. 7, no. 3, pp. 38–41, May 2016, doi: 10.1109/MPUL.2016.2539105.

[38] S. Pinto, J. Cabral, and T. Gomes, ‘We-care: An IoT-based health care system for elderly people’, *Proceedings of the IEEE International Conference on Industrial Technology*, pp. 1378–1383, Apr. 2017, doi: 10.1109/ICIT.2017.7915565.

[39] H. Suzuki *et al.*, ‘An Updated Watch-Over System Using an IoT Device, for Elderly People Living by Themselves’, *Proceedings - 2018 3rd International Conference on System Reliability and Safety, ICSRS 2018*, pp. 115–119, Apr. 2019, doi: 10.1109/ICSRS.2018.8688843.

[40] B. D. Chung Hua, H. Fahmi, L. Yuhao, C. C. Kiong, and A. Harun, ‘Internet of Things (IOT) Monitoring System for Elderly’, *International Conference on Intelligent and Advanced System, ICIAS 2018*, Nov. 2018, doi: 10.1109/ICIAS.2018.8540567.

[41] A. Azizan, N. M. Din, D. R. M. Dzaki, A. D. I. A. Kadir, and N. Shafie, ‘Implementation of an IoT Cloud-based Elderly Care Health Monitoring Dashboard’, *4th International Conference on Smart Sensors and Application: Digitalization for Societal Well-Being, ICSSA 2022*, pp. 150–154, 2022, doi: 10.1109/ICSSA54161.2022.9870939.

[42] P. Araújo and R. Salvado, ‘SmartVest: Integrating clothing with domotics, health care and information technologies’, 2010.

[43] S. Ponce *et al.*, ‘Wearable sensors and domotic environment for elderly people’, in *IFMBE Proceedings*, Springer Verlag, 2019, pp. 195–200. doi: 10.1007/978-981-10-9023-3_35.

[44] P. S. Sundaram, N. Basker, and N. L., ‘Smart clothes with bio-sensors for ECG monitoring’, *International Journal of Advanced Technology and Engineering Exploration*, vol. 8, pp. 298–301, Aug. 2019, Accessed: Aug. 16, 2023. [Online]. Available:

https://www.researchgate.net/publication/331331285_Smart_clothes_with_bio-sensors_for_ECG_monitoring

[45] S. and I. P. A. Dupre Anthony and Vincent, ‘Basic ECG Theory, Recordings, and Interpretation’, in *Handbook of Cardiac Anatomy, Physiology, and Devices*, P. A. Iaizzo, Ed., Totowa, NJ: Humana Press, 2005, pp. 191–201. doi: 10.1007/978-1-59259-835-9_15.

[46] V. Thambawita *et al.*, ‘DeepFake electrocardiograms using generative adversarial networks are the beginning of the end for privacy issues in medicine’, *Sci Rep*, vol. 11, no. 1, pp. 1–8, 2021.

[47] Wikipedia contributors, ‘Electrocardiography — Wikipedia, The Free Encyclopedia’. 2023. [Online]. Available:

<https://en.wikipedia.org/w/index.php?title=Electrocardiography&oldid=1153483880>

[48] G. B. Moody and R. G. Mark, 'The impact of the MIT-BIH Arrhythmia Database', *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, May 2001, doi: 10.1109/51.932724.

[49] S. H. Jambukia, V. K. Dabhi, and H. B. Prajapati, 'Classification of ECG signals using machine learning techniques: A survey', in *2015 International Conference on Advances in Computer Engineering and Applications*, Mar. 2015, pp. 714–721. doi: 10.1109/ICACEA.2015.7164783.

[50] M. Moavenian and H. Khorrami, 'A qualitative comparison of Artificial Neural Networks and Support Vector Machines in ECG arrhythmias classification', *Expert Syst Appl*, vol. 37, no. 4, pp. 3088–3093, 2010, doi: <https://doi.org/10.1016/j.eswa.2009.09.021>.

[51] V. S. R. Kumari and P. R. Kumar, 'Cardiac arrhythmia prediction using improved multilayer perceptron neural network', *International Journal of Electronics, Communication & Instrumentation Engineering Research and Development (IJECIERD)*, vol. 3, no. 4, pp. 73–80, 2013, Accessed: Jul. 04, 2023. [Online]. Available:

https://d1wqtxts1xzle7.cloudfront.net/32156629/8._Electronics_-_IJECIERD_-_CARDIAC_-_VSR_Kumari_-_Full-libre.pdf?1391533094=&response-content-disposition=inline%3B+filename%3DCardiac_Arrhythmia_Prediction_Using_Impr.pdf&Expires=1688466625&Signature=TXdEcsqBkRsNqweeyAJ-P61SmaPgYvvYSjvLzkDObAcsqT9kFAL5ngEYbwByYSZax-uiDK76QE1nB3~YVzcvFi-zU9yz9rg~UKzzqqorw4JotKKGqeIIKndd15mA~AcCTqEM2soN7LJDsEh2Ngv05jN~Ts5gygBb3Nz4E1mtith6WyvAMp4mtMKysYqgMlZzAPtlvGA6Y8dQ0KLOixJTekF3S1fi~ujqUC6eI8Sei2nVk48t3Johue9vyNVmQYuea~4l-HaavOnvUE~lutduHIQT0bFjI7R~gq0Co6kyAUYi-gMQTIkaUJEpNV~qhW53XP3c0Fy4nhYUI589Bukf86w__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[52] D. Patra, M. Das, and S. Pradhan, 'Integration Of Fcm, Pca And Neural Networks For Classification Of Ecg Arrhythmias', *LAENG Int J Comput Sci*, vol. 36, Jul. 2009, Accessed: Jul. 04, 2023. [Online]. Available: https://www.researchgate.net/publication/26843936_Integration_Of_Fcm_Pca_And_Neural_Networks_For_Classification_Of_Ecg_Arrhythmias

[53] S. Kaplan Berkaya, A. K. Uysal, E. Sora Gunal, S. Ergin, S. Gunal, and M. B. Gulmezoglu, 'A survey on ECG analysis', *Biomed Signal Process Control*, vol. 43, pp. 216–235, 2018, doi: <https://doi.org/10.1016/j.bspc.2018.03.003>.

[54] R. Bousseljot, D. Kreiseler, and A. Schnabel, 'Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet', 1995. Accessed:

- Jul. 05, 2023. [Online]. Available: <https://www.deepdyve.com/lp/de-gruyter/nutzung-der-ekg-signaldatenbank-cardiodat-der-ptb-ber-das-internet-uemKpjIFzM>
- [55] A. H. Ribeiro *et al.*, ‘Automatic diagnosis of the 12-lead ECG using a deep neural network’, *Nat Commun*, vol. 11, no. 1, Dec. 2020, doi: 10.1038/s41467-020-15432-4.
- [56] A. H. Ribeiro *et al.*, ‘CODE-15%: a large scale annotated dataset of 12-lead ECGs’, Jun. 2021, doi: 10.5281/ZENODO.4916206.
- [57] L. Xiaolin, W. Qingyuan, R. C. Panicker, B. Cardiff, and D. John, ‘Binary ECG Classification Using Explainable Boosting Machines for IoT Edge Devices’, in *2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2022, pp. 1–4. doi: 10.1109/ICECS202256217.2022.9970834.
- [58] J. Zheng *et al.*, ‘Optimal Multi-Stage Arrhythmia Classification Approach’, *Sci Rep*, vol. 10, no. 1, p. 2898, 2020, doi: 10.1038/s41598-020-59821-7.
- [59] P. Wagner *et al.*, ‘PTB-XL, a large publicly available electrocardiography dataset’, *Sci Data*, vol. 7, no. 1, p. 154, 2020, doi: 10.1038/s41597-020-0495-6.
- [60] A. L. Goldberger *et al.*, ‘PhysioBank, PhysioToolkit, and PhysioNet’, *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000, doi: 10.1161/01.CIR.101.23.e215.
- [61] PhysioNet, ‘WFDB Software Package’. PhysioNet, 2023. Accessed: Jul. 06, 2023. [Online]. Available: <https://archive.physionet.org/physiotools/wfdb.shtml>
- [62] The HDF Group, ‘Hierarchical Data Format, version 5’. The HDF Group, 2023. Accessed: Jul. 06, 2023. [Online]. Available: <https://www.hdfgroup.org/HDF5/>
- [63] E. Ajdaraga and M. Gusev, ‘Analysis of sampling frequency and resolution in ECG signals’, in *2017 25th Telecommunication Forum (TELFOR)*, 2017, pp. 1–4. doi: 10.1109/TELFOR.2017.8249438.
- [64] P. Virtanen *et al.*, ‘SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python’, *Nat Methods*, vol. 17, pp. 261–272, 2020, doi: 10.1038/s41592-019-0686-2.
- [65] C. Carreiras, A. Alves, A. Lourenço, F. Caneto, H. Silva, and A. Fred, ‘BioSPPy: Biosignal Processing in Python’. 2023. Accessed: Jul. 06, 2023. [Online]. Available: <https://github.com/PIA-Group/BioSPPy/>
- [66] R. J. Martis, U. R. Acharya, H. Prasad, C. K. Chua, C. M. Lim, and J. S. Suri, ‘Application of higher order statistics for atrial arrhythmia

classification’, *Biomed Signal Process Control*, vol. 8, no. 6, pp. 888–900, 2013, doi: <https://doi.org/10.1016/j.bspc.2013.08.008>.

[67] Y. Kutlu and D. Kuntalp, ‘A multi-stage automatic arrhythmia recognition and classification system’, *Comput Biol Med*, vol. 41, no. 1, pp. 37–45, 2011, doi: <https://doi.org/10.1016/j.combiomed.2010.11.003>.

[68] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. Accessed: Jul. 11, 2023. [Online]. Available: <https://docs.python.org/3/reference/>

[69] T. Kluyver *et al.*, ‘Jupyter Notebooks – a publishing format for reproducible computational workflows’, in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Schmidt, Eds., 2016, pp. 87–90. Accessed: Jul. 11, 2023. [Online]. Available: <https://ebooks.iospress.nl/publication/42900>

[70] F. Pedregosa *et al.*, ‘Scikit-learn: Machine Learning in Python’, *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011, Accessed: Jul. 06, 2023. [Online]. Available: <https://scikit-learn.org/stable/about.html#citing-scikit-learn>

[71] Martⁿ Abadi *et al.*, ‘TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems’. 2015. [Online]. Available: <https://www.tensorflow.org/>

[72] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, ‘A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data’, *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, Jun. 2004, doi: 10.1145/1007730.1007735.

[73] W. Sousa Lima, E. Souto, K. El-Khatib, R. Jalali, and J. Gama, ‘Human Activity Recognition Using Inertial Sensors in a Smartphone: An Overview’, *Sensors*, vol. 19, no. 14, 2019, doi: 10.3390/s19143213.

[74] D. Garcia-Gonzalez, D. Rivero, E. Fernandez-Blanco, and M. R. Luaces, ‘A Public Domain Dataset for Real-Life Human Activity Recognition Using Smartphone Sensors’, *Sensors*, vol. 20, no. 8, 2020, doi: 10.3390/s20082200.

[75] C. Dewi and R.-C. Chen, ‘Human Activity Recognition Based on Evolution of Features Selection and Random Forest’, in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Oct. 2019, pp. 2496–2501. doi: 10.1109/SMC.2019.8913868.

[76] ‘Motion sensors | Android Developers’. Accessed: Jul. 17, 2023. [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_motion

- [77] 'Location | Android Developers'. Accessed: Jul. 17, 2023. [Online]. Available: <https://developer.android.com/reference/android/location/Location>
- [78] R. Khusainov, D. Azzi, I. E. Achumba, and S. D. Bersch, 'Real-Time Human Ambulation, Activity, and Physiological Monitoring: Taxonomy of Issues, Techniques, Applications, Challenges and Limitations', *Sensors 2013, Vol. 13, Pages 12852-12902*, vol. 13, no. 10, pp. 12852–12902, Sep. 2013, doi: 10.3390/S131012852.
- [79] 'VS_PRODUCT_VSH101 – VitalSigns Technology'. Accessed: Jul. 25, 2023. [Online]. Available: https://www.vsigntek.com/vs_product_vsh101/
- [80] Alexander Shvets, *Dive Into Design Patterns*. Alexander Shvets, 2019. Accessed: Jul. 26, 2023. [Online]. Available: <https://refactoring.guru/design-patterns/book>

APPENDIX

Appendix A – Internship proposal



Master in Computer Science Internship proposal

Delete what isn't applicable

The academic year of 2022/2023

Title	IoT Platform for Personal Healthcare of Elderly People
Student	Ruben Marques
Supervisor (School)	Francisco Pereira, PhD, xico@isec.pt
Company	Kirepo S.à.r.l
Supervisor (Company)	Steve Reisdorf – CEO – steve@kirepo.lu
Workplace	Luxembourg / Mamer / Headquarter Kirepo & Remotely

SUMMARY

Kirepo is in the process of developing an IoT-based platform to support elderly people in staying longer in their houses without external daily help. The platform, through connected devices, creates profiles and evaluates exceptions which will then, if a threshold has been reached, trigger a call centre that will either contact the person directly or/and send professional help.

The challenge of this platform is to communicate with IoT sensors to identify the status of the person, to interact with the person (if 1 issue is found, automatically ask other sensors to verify), and pilot other devices (open doors, allow a person-to-person dialogue). All those sensors should allow the creation of a profile of the person's daily routine, which is then analysed by artificial intelligence to detect anomalies and automate workflows.

1. SCOPE

This platform is considered as state of the art, innovative, and composes several core computer science topics (IoT interactions, APIs, network, centralized platform, AI, triggers). The concept of the platform has been presented to healthcare associations which clearly support the idea and the concept and they're willing to use it and are also happy to help during development. Other suppliers/installation partners are also attached to the concept. The project is currently focused on Luxembourg due to its size, but it is, of course, fully expandable to other countries. It is also not limited to elder people but can be generalized.

2. GOALS

This internship aims to reach the following generic goals:

- Create an inventory of sensors, partner company will build a show-room to have the system's setup
- Build a platform which will read out several off-the-shelf sensors and collect / store/ analyse their data



- Build a level of artificial intelligence which will read this data and create a pattern where exceptions are filtered out and leading to triggers
- Based on triggers create roles to access/start other sensors to confirm/reject the initial finding. Create a level system, where the handover to a physical person will be initiated if the value is above the threshold.
- Extract the data in a human-readable way and send it to the human operator as a basis for the upcoming contact with the person.
- Document the project allowing other persons to co-work / take over the project at the end.

3. WORK SCHEDULE

Note: The total workload consists of about 1170 hours: 390 hours in the 1st semester and 780 hours in the 2nd semester.

The internship consists of the following activities and respective tasks:

T1 – Introduction of the student to the existing work (mainly conceptual), but also set up introductions to the hardware devices already present. Define together the scope of the project as well as the milestones.

T2 – Familiarization with the home-assistant platform, an intermediate building block of the whole project. Connect devices to the platform, evaluate the API endpoints and elaborate a communication logic between the home-assistant devices and the central server

T3 – Set up a centralized server and services to accept home-assistant data. Create a logic of a data lake to gather all incoming data to allow pattern creation to feed the AI

T4 – Data analysis with AI logic in mind to allow automatic detection of abnormal patterns in the incoming data. The outcome should be a trigger-based system which reacts automatically to the gathered data.

T5 – Create a user-faced interface (dashboard, email sending, webhooks, ...) which will be triggered by the underlying systems.

T6 – Creating documentation of the different phases.

The following documents and materials will be produced:

D1 – Platform overview, from the device at home to the centralized server infrastructure where the professionals are connected to

D2 – MVP client-side – Initial prototype of connected devices sending data to a central server

D3 – MVP server-side - Initial prototype of a centralized server receiving data

D4 – MVP data analysis/AI - Initial prototype of a model which analyses data and detects abnormalities

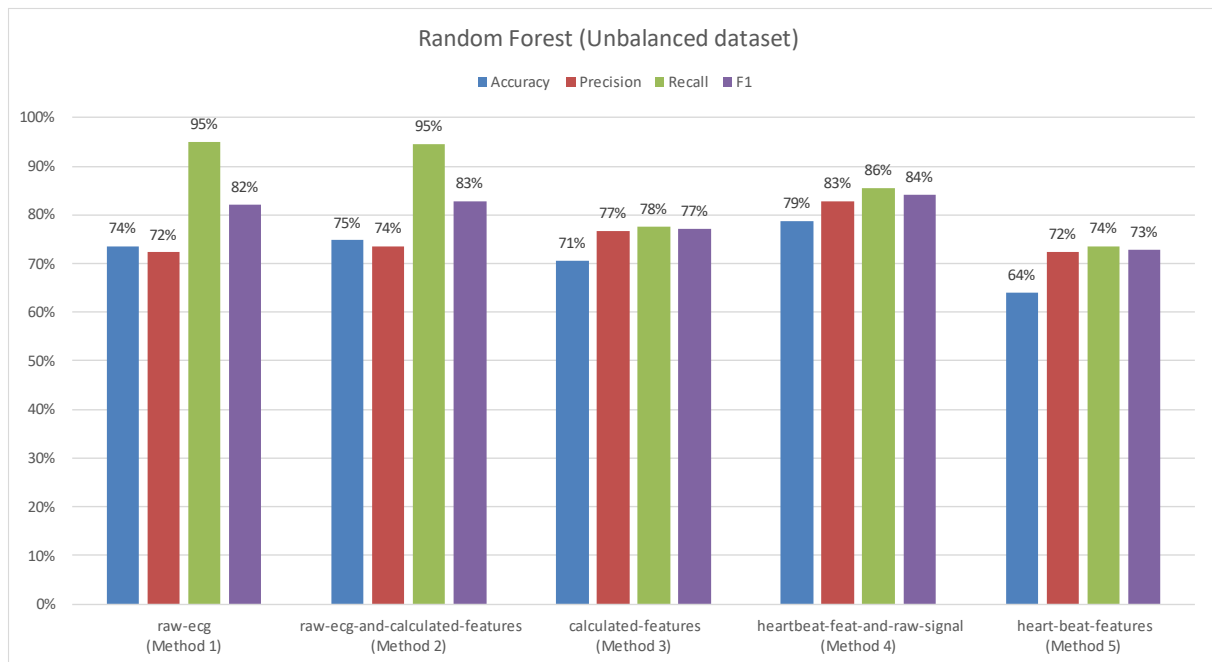
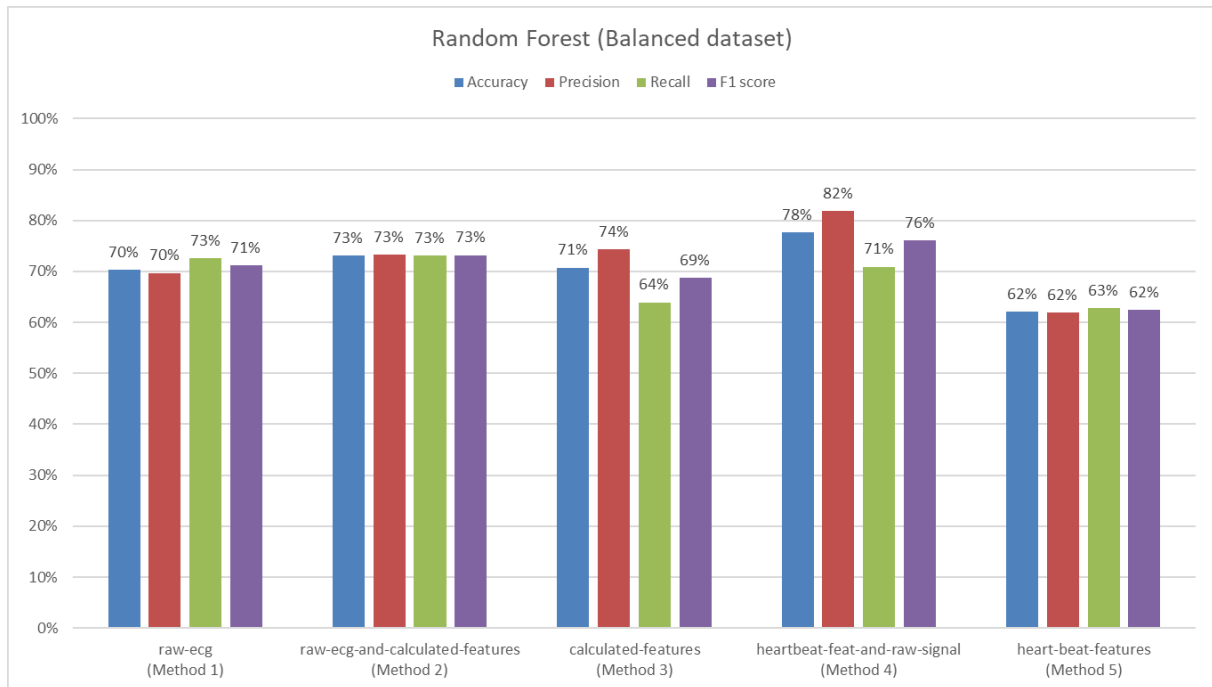
Dx – Summary for the mid-term workshop – Summary and PPT presentation for the mid-term workshop (somewhere in between D2 and D4, based on the progress (as D2 to D5 can and will probably be initiated in parallel.

Dn – Final report – Final report for public auditions.

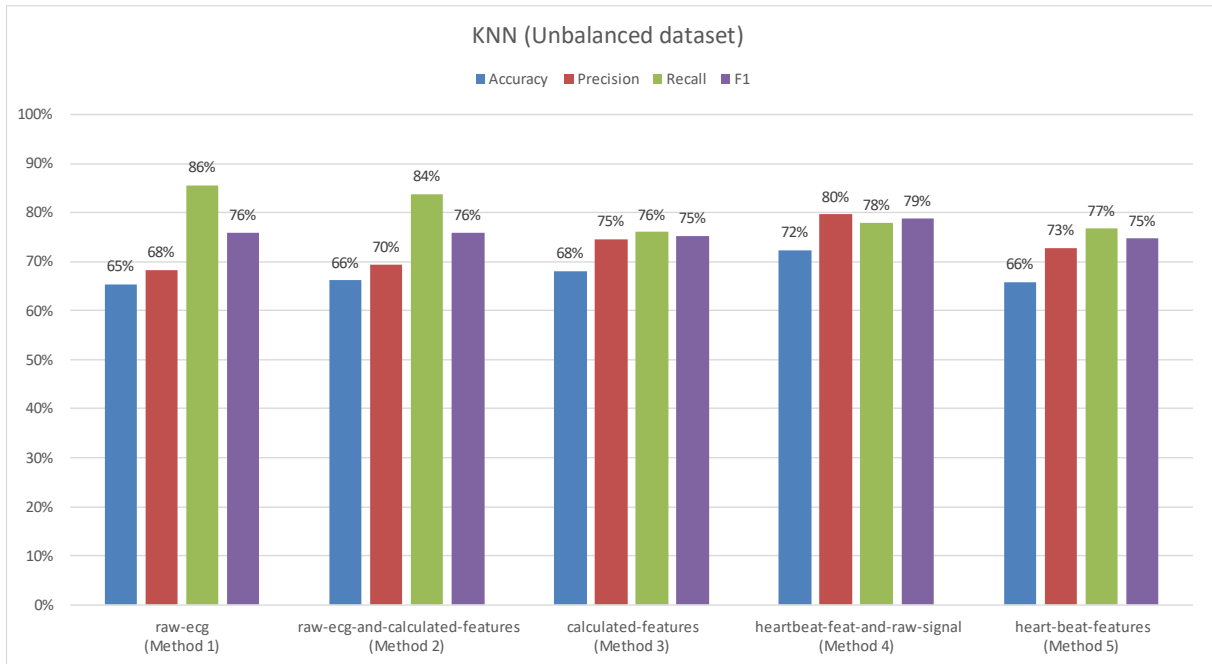
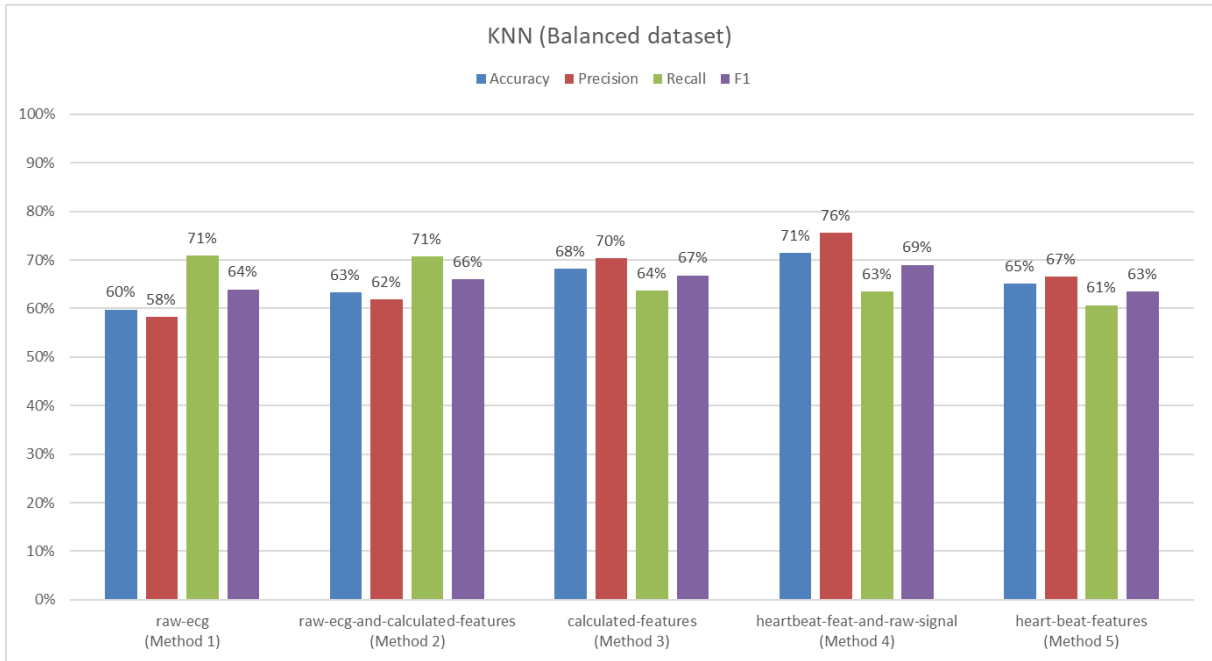
The expected schedule for the work tasks and deliveries is the following:

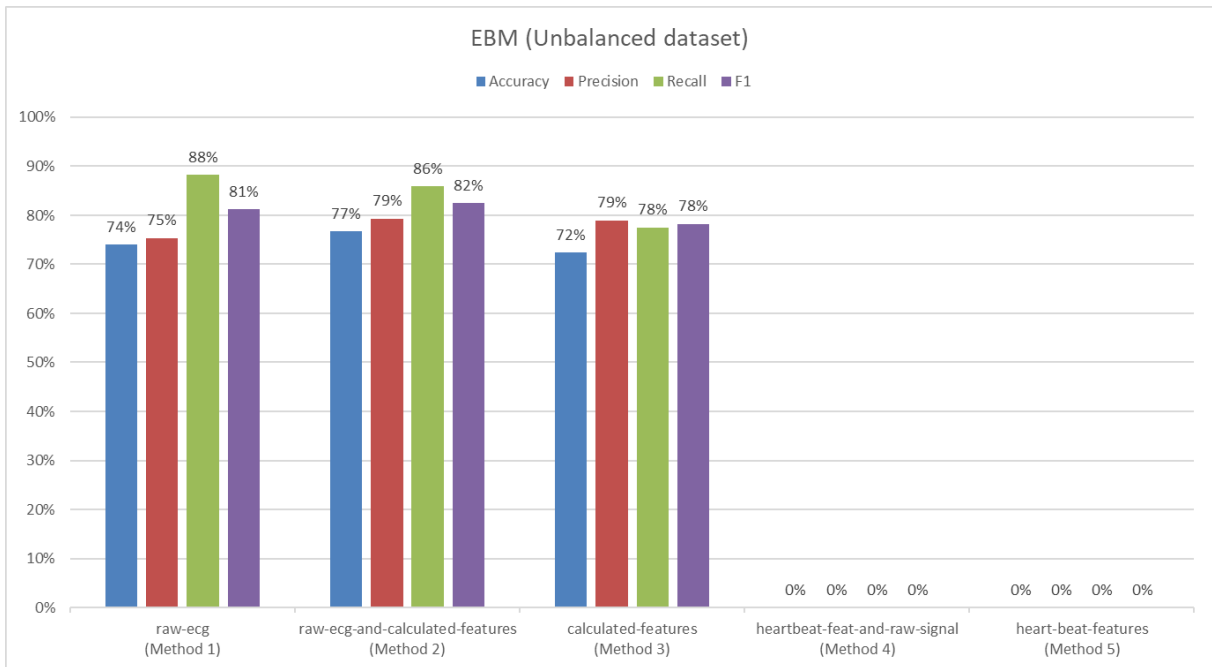
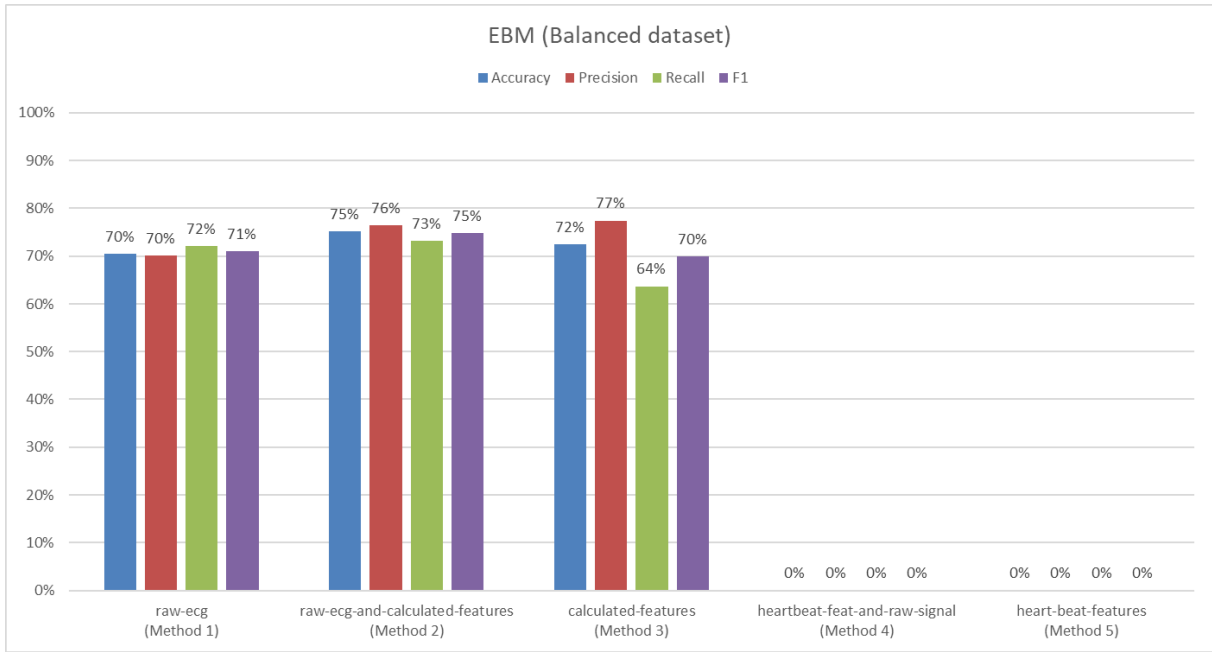
	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul
Tasks											

Appendix B – ML models’ performance charts of all ML test instances for ECG classification (Feature extraction methods comparison)

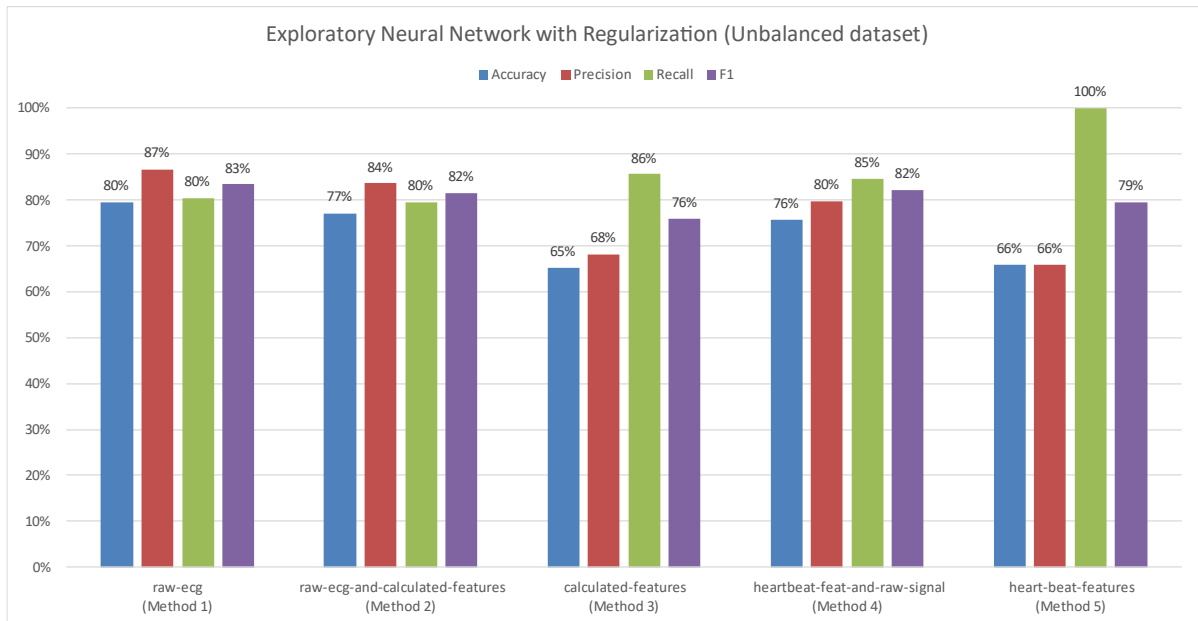
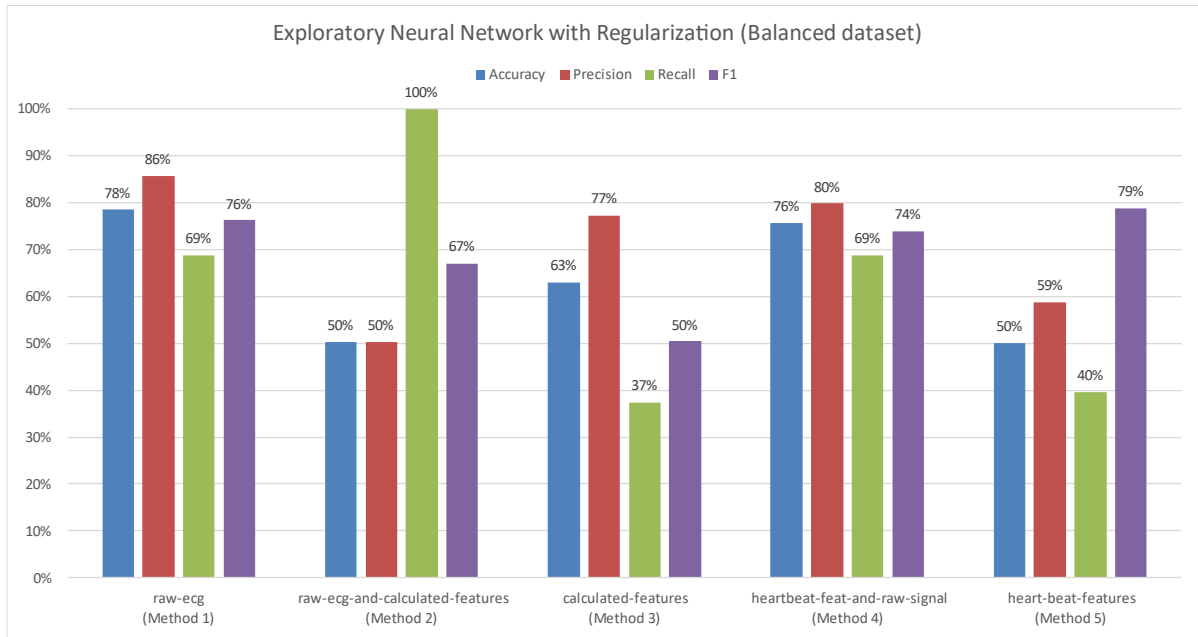


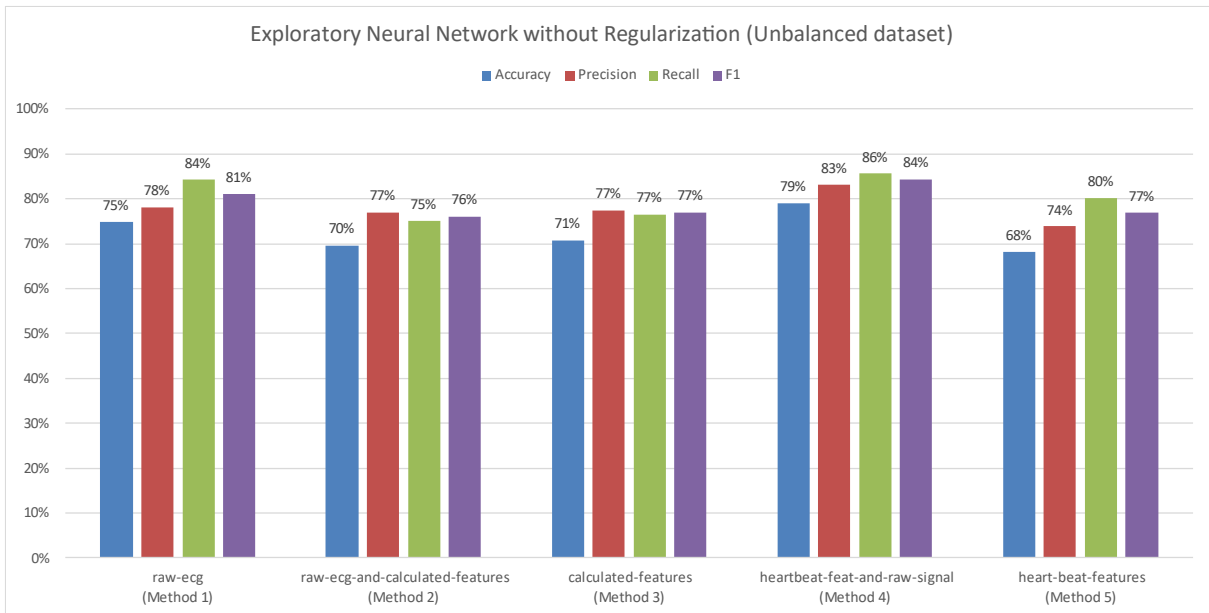
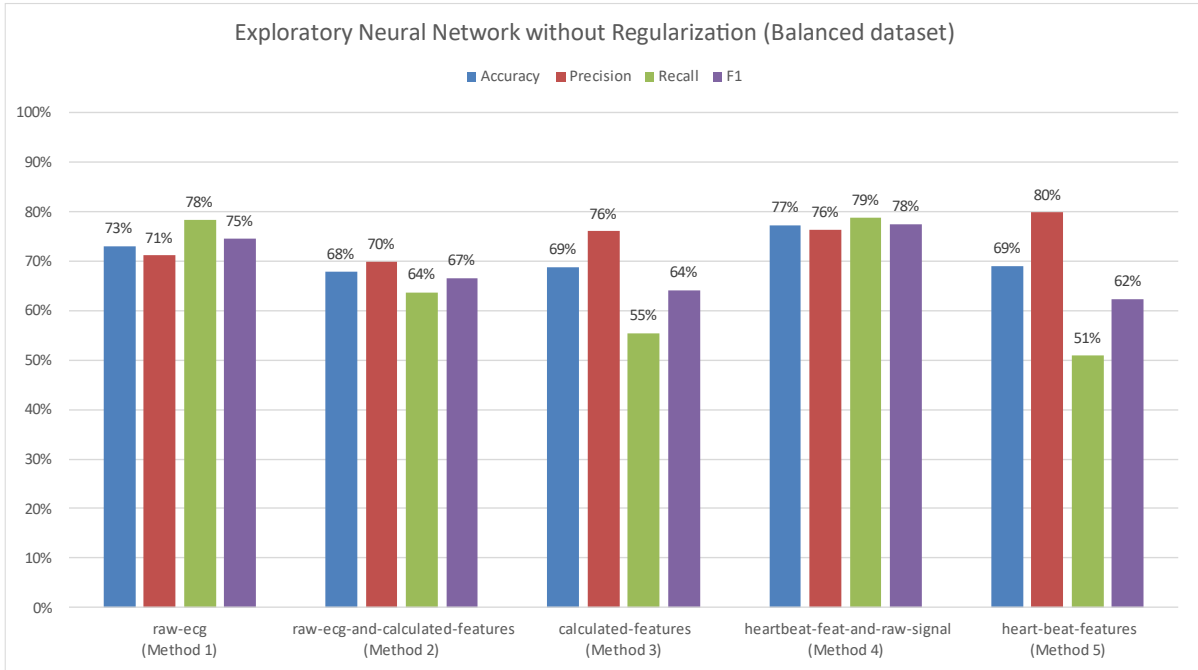
IoT Platform for Personal Healthcare of Elderly People



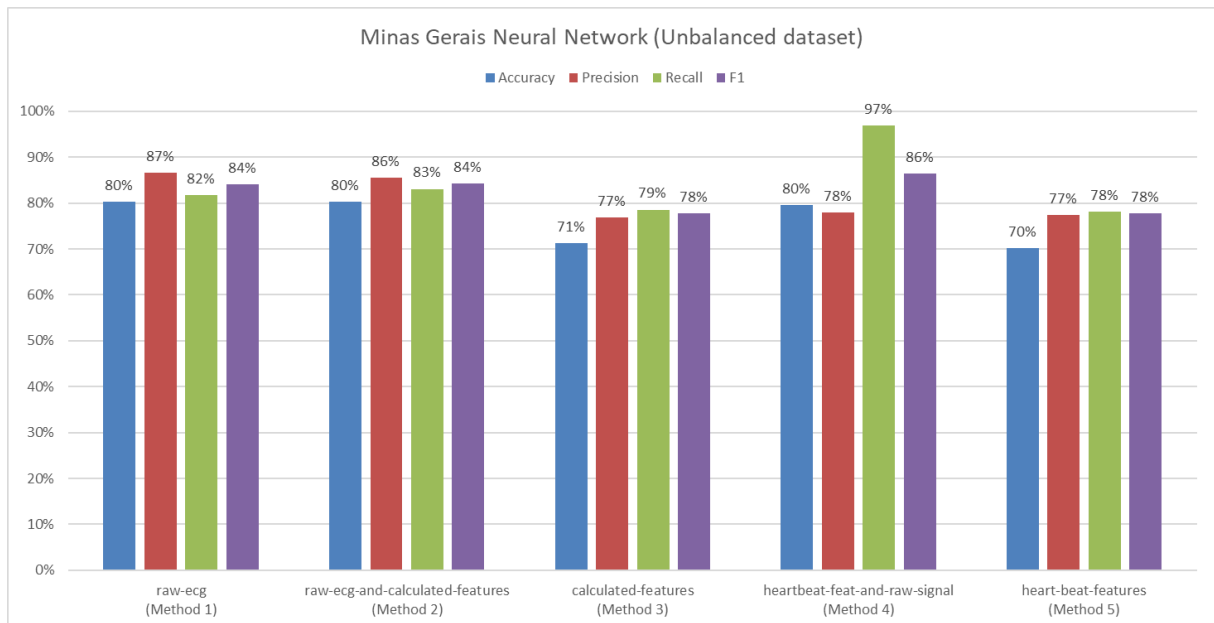
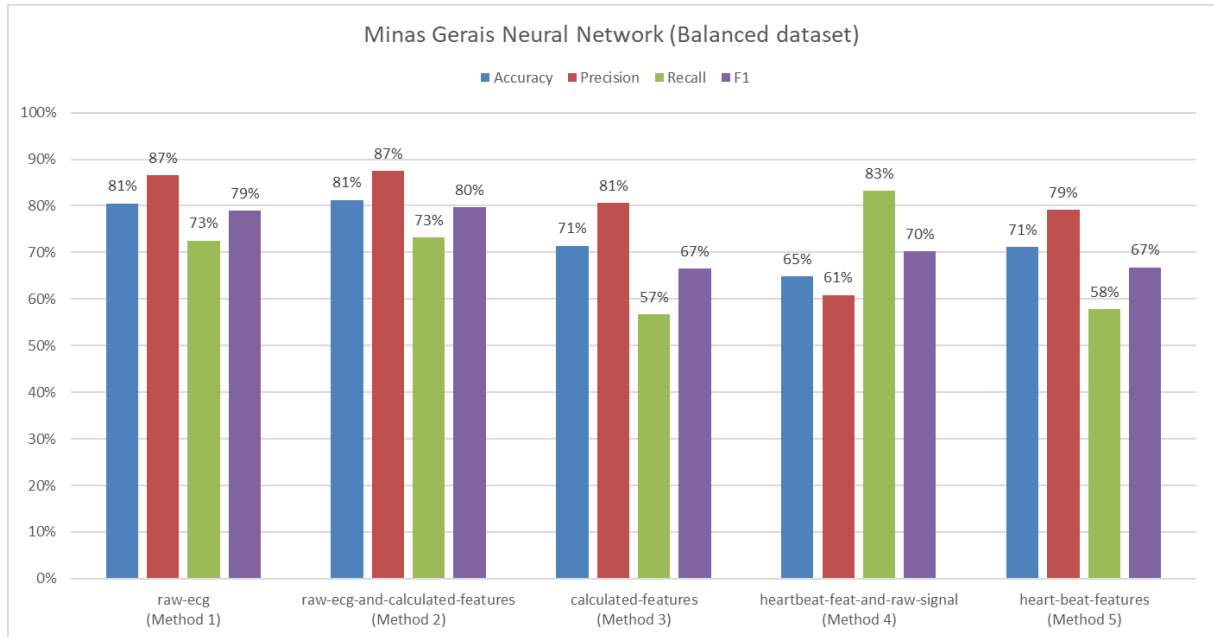


IoT Platform for Personal Healthcare of Elderly People

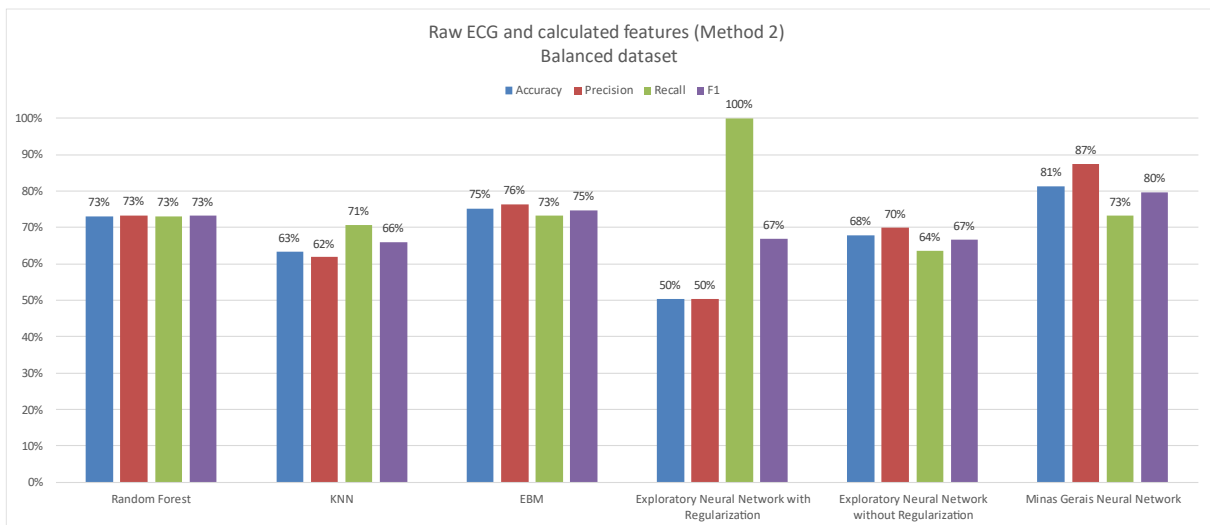
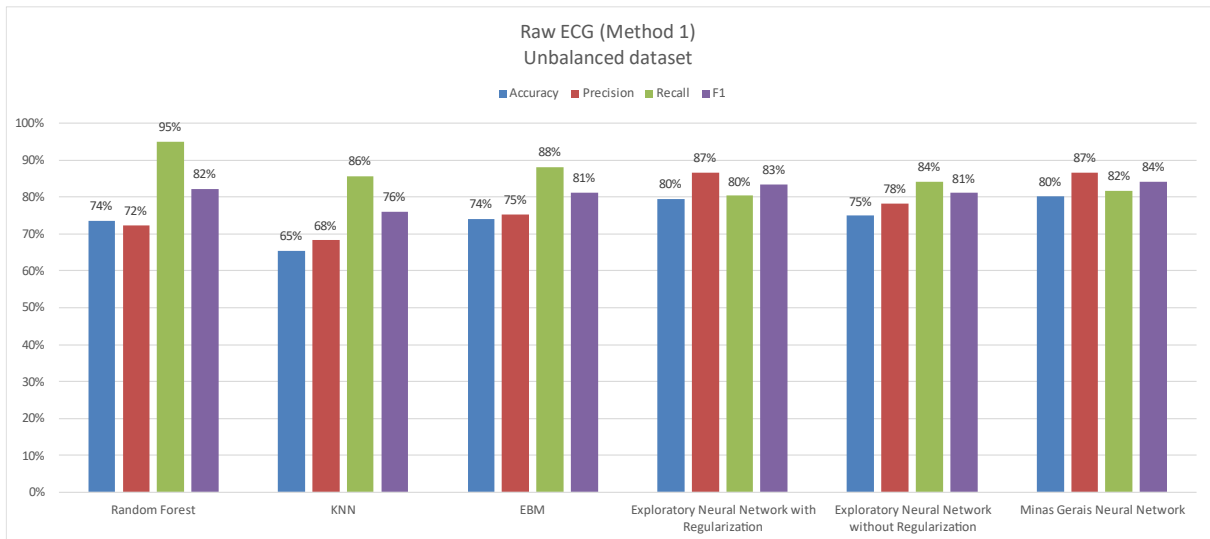
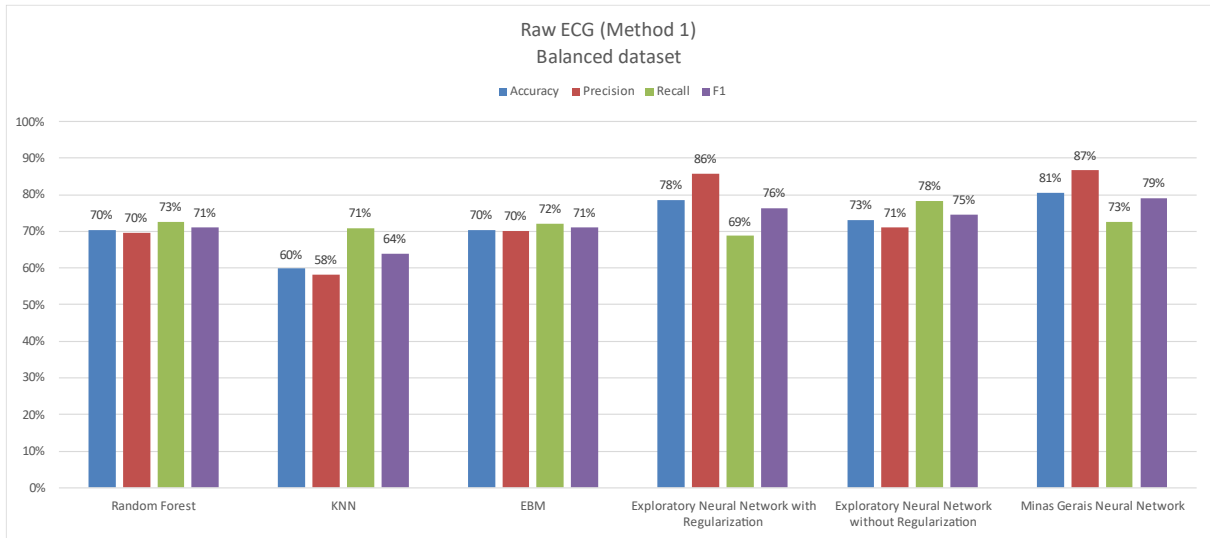




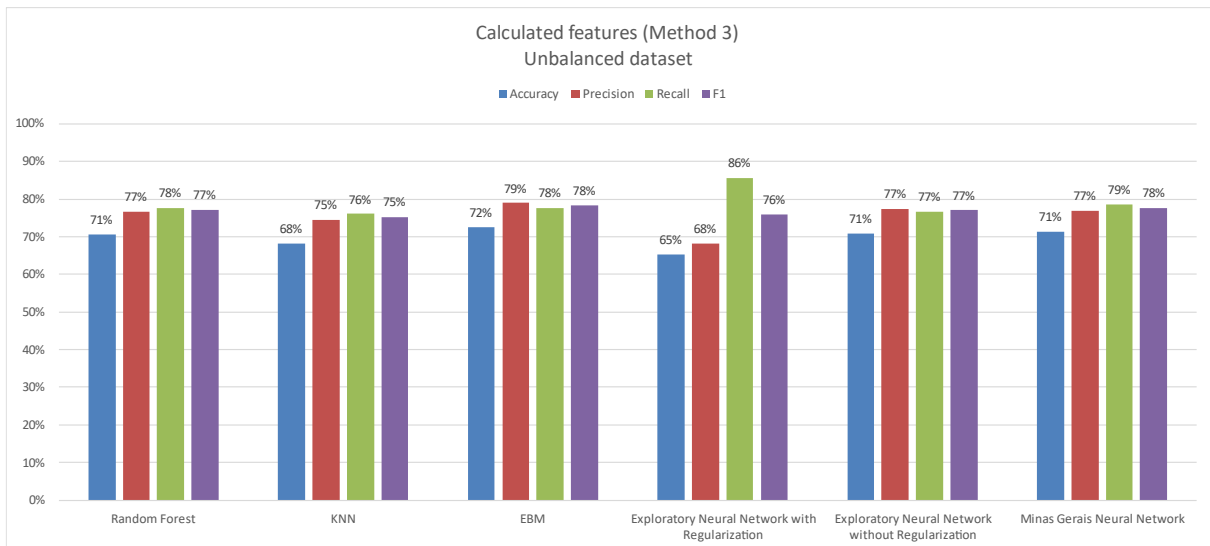
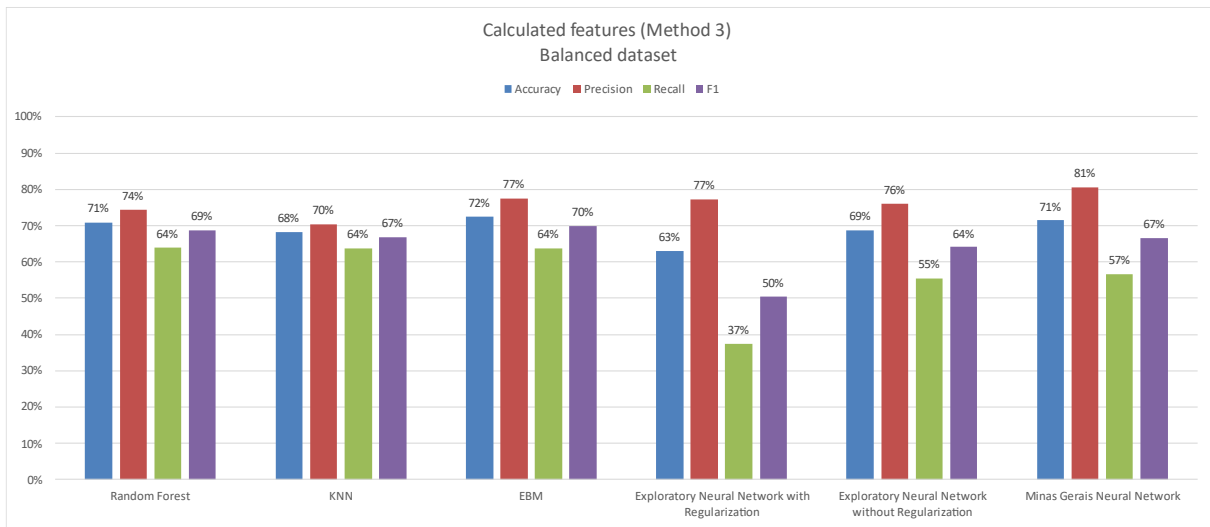
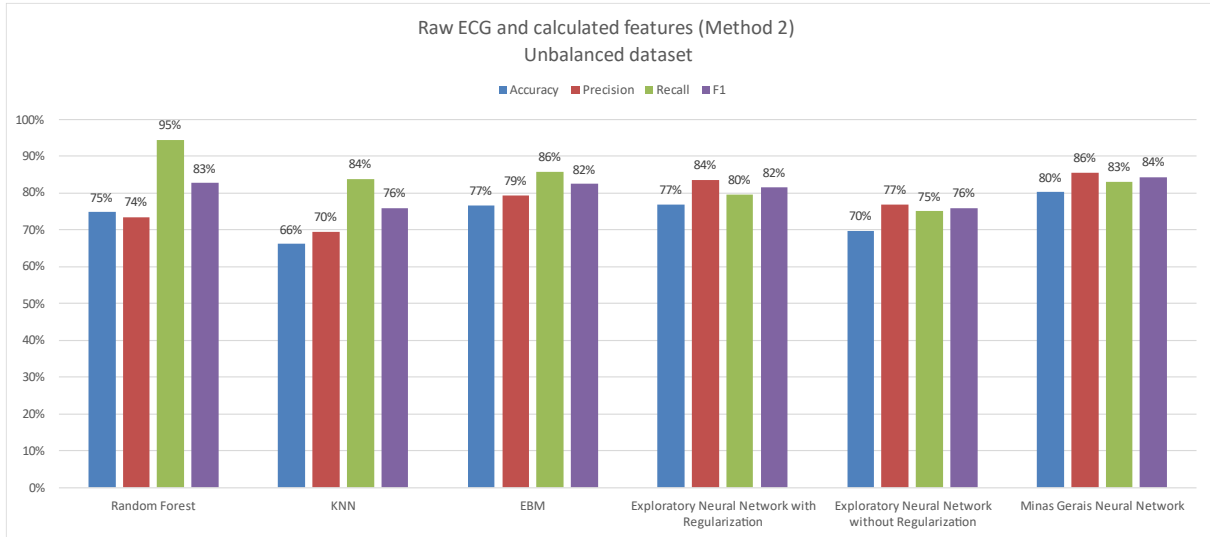
IoT Platform for Personal Healthcare of Elderly People

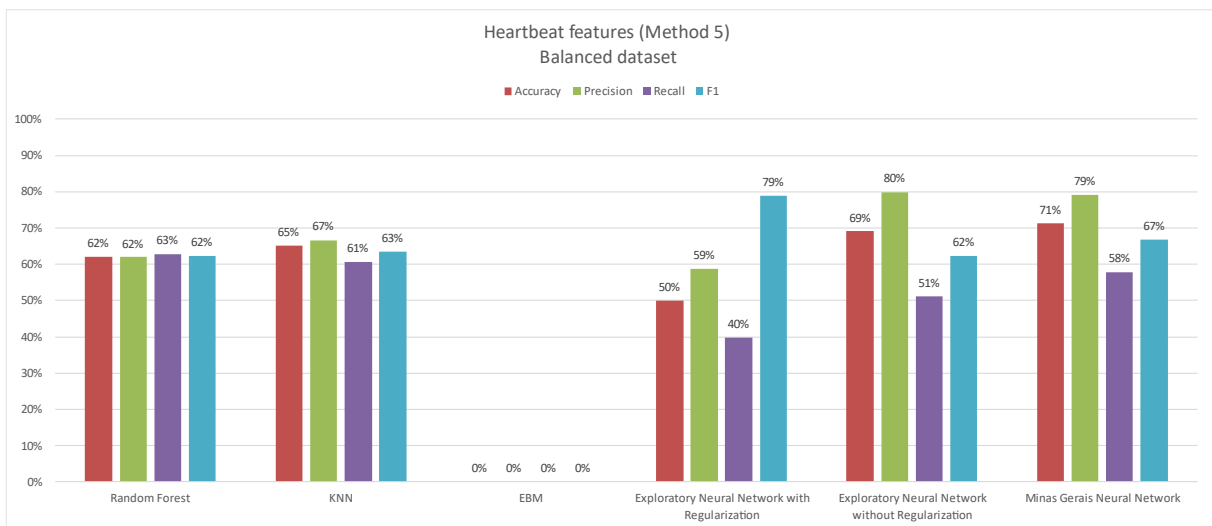
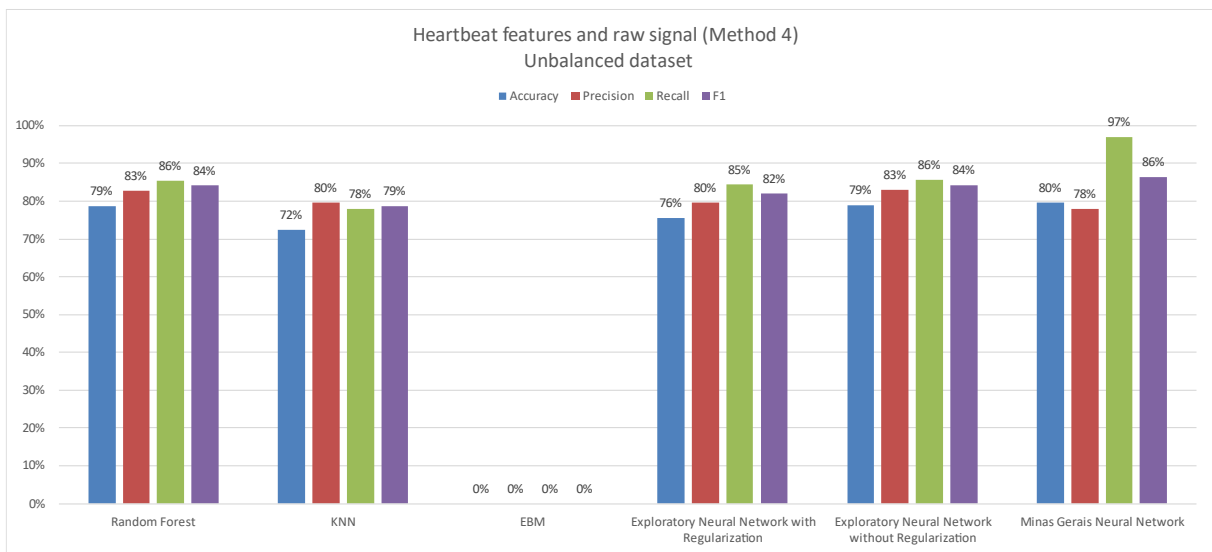
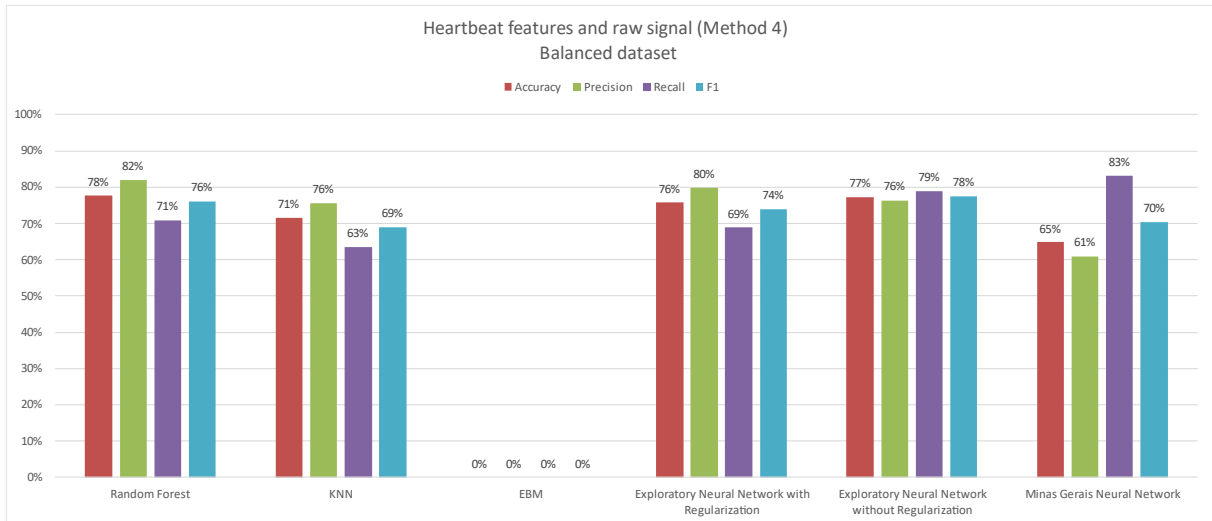


Appendix C – ML models' performance charts of all ML test instances for ECG classification (Model comparison)

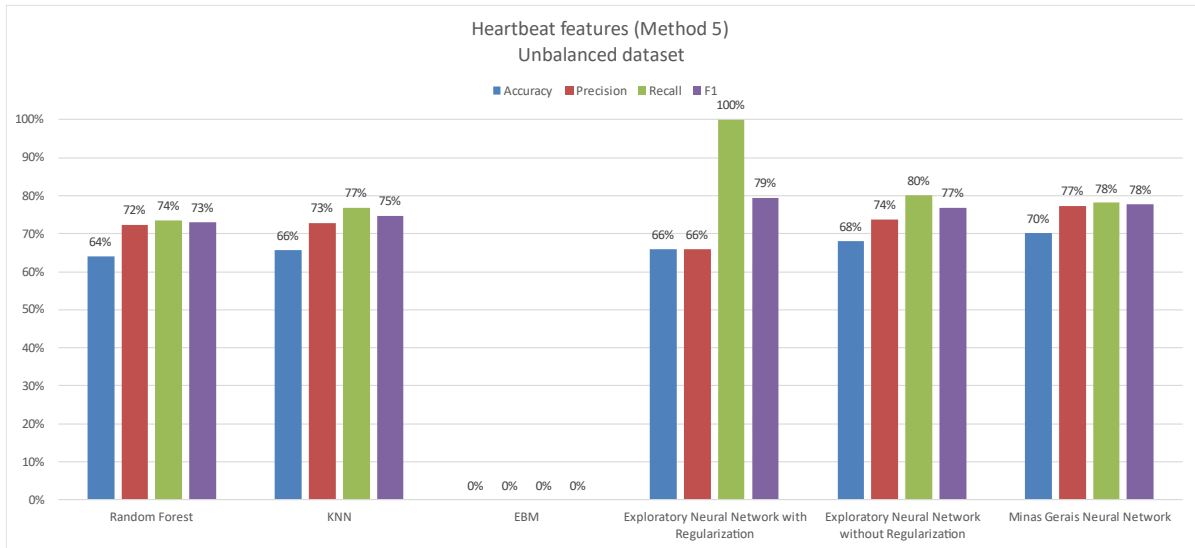


IoT Platform for Personal Healthcare of Elderly People

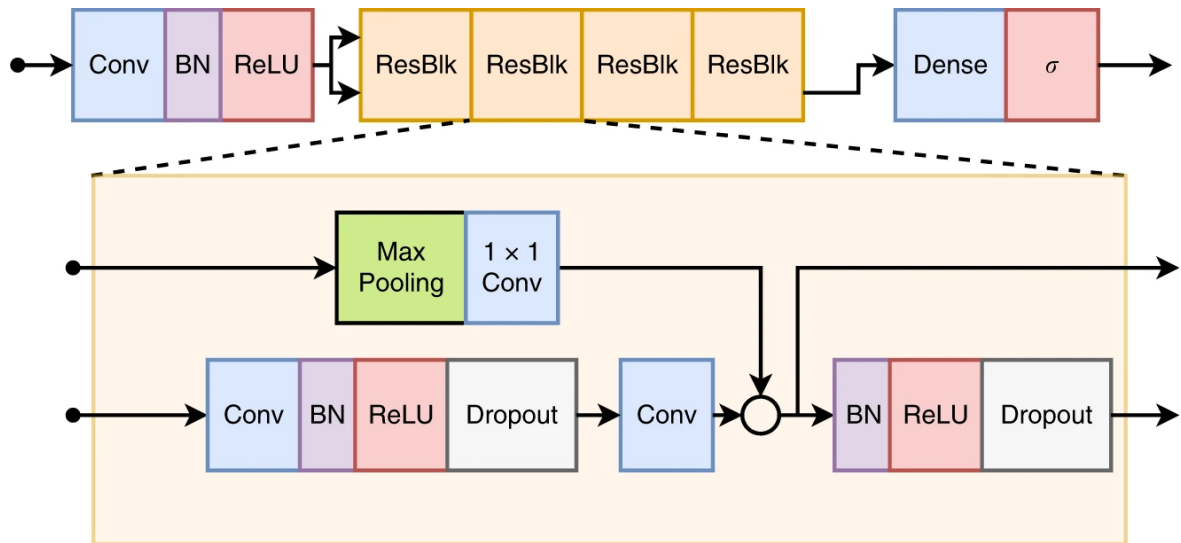




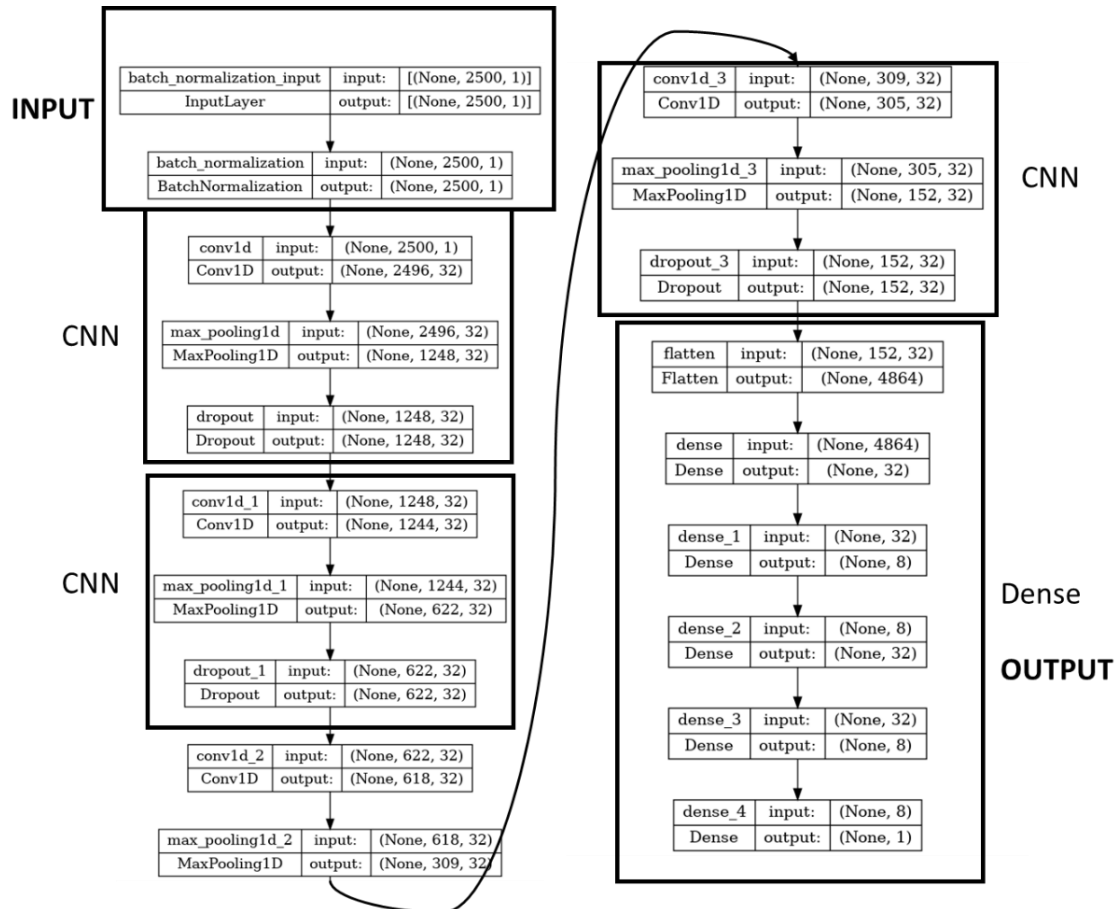
IoT Platform for Personal Healthcare of Elderly People



Appendix D – Architecture of the Minas Gerais Neural Network



Appendix E – Architecture of the Exploratory Neural Network



```

1 filters = 32
2
3 kernel_size = 5
4
5 model = Sequential()
6 model.add(BatchNormalization(input_shape=(number_of_features,1)))
7 # number_of_features = Number of feature columns in the DataFrame
8 model.add(Conv1D(filters=filters, kernel_size=kernel_size, activation='relu'))
9 model.add(MaxPooling1D())
10 model.add(Dropout(0.2))
11
12 for _ in range(3):
13     model.add(Conv1D(filters=filters, kernel_size=kernel_size, activation='relu'))
14     model.add(MaxPooling1D())
15     model.add(Dropout(0.2))
16
17 model.add(Flatten())
18
19 for _ in range(2):
20     model.add(Dense(32, activation="relu"))
21     model.add(Dense(8, activation="relu"))
22
23 model.add(Dense(1, activation="sigmoid"))
24
25 loss = 'binary_crossentropy'
26 lr = 0.001
27 opt = Adam(lr)
28 model.compile(loss=loss, optimizer=opt, metrics=['accuracy'])

```




**Instituto Superior
de Engenharia**

Politécnico de Coimbra