Gennaro Laudato

Innovative Information Systems to Monitor Biomedical Parameters during High Demanding Tasks

PhD Thesis
University of Molise, Italy
April, 2021

Università degli Studi del Molise

Dipartimento di Bioscienze e Territorio

Corso di Dottorato in Bioscienze e Territorio
Ciclo XXXIII
S.S.D. ING-INF/05

PhD Thesis

Innovative Information Systems to Monitor Biomedical Parameters during High Demanding Tasks

Coordinatore
Chiar.mo Prof. Giovanni Fabbrocino

Relatore/Tutor Accademico
Chiar.mo Prof. Rocco Oliveto

Candidato
Gennaro Laudato

Anno Accademico 2019/2020

Abstract

The objective of this PhD project is, as its research core, the application of Machine Learning techniques and Big Data analytics to monitor, in a non-invasive way, vital parameters of individuals engaged in tasks that require a high psychophysical effort. The industrial partners of this project are Formula Medicine (as Italian industrial partner with advisor Dr. Riccardo Ceccarelli) and AOTech (foreign industrial partner with advisor mr. Sebastien Philippe). Formula Medicine is a sports medicine center able to offer medical assistance and training programs both physical and mental. Its strength is represented by the Mental Economy Gym, a gym dedicated to the optimization of mental resources. AOTech, on the other hand, is a partner company of Formula Medicine and a leader in the definition of high-tech products and services for the automotive industry and motorsport. AOTech has designed a sport driving simulator, able to reproduce all the main world circuits. The simulator, thanks to the equipment of a hydraulic system, allows to relive physical and mental sensations very similar to those perceived during real driving. The software system also allows vehicle's data extraction. Within the present project, also taking into account the research domain of the industrial partners, the focus has been addressed to the monitoring of athletes belonging to motorsport with two linked but distinct objectives: the first, strictly related to the analysis of the drivers' body performances and the second dedicated to the automatic identification of cardiac pathologies starting from electrocardiographic data. Finally, the know-how on the monitoring of biomedical parameters, acquired during the first years of this PhD project in the field of motorsport, was exported to the field of software engineering with the aim of verifying the possibility of predicting the correctness of a programming task that a software developer performs, based on the continuous monitoring of his body parameters. As a first result of the PhD, novel metrics have been defined to objectify effort, *physical consumption*, stress, and other factors. These metrics have been included in the software in use in Formula Medicine to have a measure of performance. In addition, part of them were correlated with the race

performance of the drivers, through the integration of the body data with the data derived from the driving simulator used in AOTech. With regards to the second research focus, a decision support system was defined in the context of early diagnosis of cardiac diseases. The recommendation system consists of several algorithms that accept as input a digital electrocardiographic lead and identify the presence of a possible cardiac pathology.

Finally, in the software engineering research field, the production of a developer was measured by evaluating the absence of defects in the source code. Preliminary results show that the proposed approach—that takes into account biomedical and code-based features—allows to discriminate with fair accuracy the outcome of a programming task, reaching an accuracy higher than 80%. This result was compared to state of the art metrics based on measures on the source code. It was higher than source code metrics, thus demonstrating the importance of biometric measurements in the identification of correctness of a coding task.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Contents

Biometric monitoring is spreading among a large variety of applications. Sports is one of the most common fields of application. Sports is an industry of high stakes. The world of professional competition is always trying to improve the athletic efficiency of its working athletes, who must compete under demanding circumstances, including exhaustion, overwork, overtraining, and sleep deprivation. These raise the risk of injuries and damage to soft tissue, the biggest risks to playoff chances. Such cases are not only unfortunate; they are highly expensive. Indeed, the reason is also economic with athlete contracts costing tens of millions of dollars and prestigious competitions at risk. Some have reported that 2.7 billion dollars were lost by the National Basketball Association (NBA) due to injuries over nine seasons, with some players losing up to 50 million a season

[224, 118].

Due to the background and research domain of the partnership, this PhD thesis is focused on motorsport athletes and it is conceptually structured in three main fields of study: the biometric monitoring in motorsport, the ECG analysis for the prevention of drivers' state of health and the biometric monitoring in software engineering. The first two sections are strictly related while the last one has been investigated only after acquiring knowledge and experience in monitoring applications. With this choice, we aimed at evaluating the potential of monitoring applications in a different context.

Therefore, the main research questions that have driven our work are the following:

> RQ_1: *Based on the biomedical monitoring data, is it possible to objectively measure the performance of a driver?*

With this RQ we aimed at answering to a request coming from trainers of professional drivers. They would have preferred to use a numeric synthesis to assess the performance of a driver during a mental test. The work dedicated to this RQ initially concerned the definition of metrics that, based on data related to a mental test (e.g., response time, score, difficulty) and biomedical monitoring data, should describe in an objective and numerical way a performance, both in terms of results obtained and in terms of ability to control the body. As a first result of this project, we integrated such metrics in the software used by the industrial partner to provide support during the car race training sessions.

> RQ_2: *By correlating the actual performance of the driver (through race data) with biomedical data, is it possible to create a tool to support a motorsport trainer?*

With this RQ we aimed at designing an innovative Trainer Support Tool for the combined analysis of vehicle and body data during a driving performance in a simulated environment. The tool is equipped with the following functionalities: (i) acquisition of the data from the two systems involved, (ii) performing the data integration and synchronization, (iii) offering a visualization section, and

(iv) providing information derived from automatic analysis of such data. In this way, a trainer can have a complete view of how the driver is coping with each moment of a race, both in terms of vehicle parameters and in terms of control of her body.

> *RQ_3: Through continuous monitoring of the drivers' vital parameters (with particular focus on ECG) is it possible to perform automatic identification of heart diseases for early diagnosis?*

With this RQ we aimed at improving the state-of-the-art methods dedicated to the automatic analysis of the ECG for the purposes of early diagnosis of pathological conditions. A Decision Support System was defined in the context of early diagnosis of cardiac diseases. The recommendation system consists of several algorithms that accept as input a digital electrocardiographic lead and identify the presence of a possible cardiac pathology. The output can be considered as an early diagnosis, supporting the specialized personnel following the driver. Initially, an innovative approach for the identification of Atrial Fibrillation (AF) has been defined that integrates features of an ECG belonging to two categories: rhythmic and morphological. These features are then provided to a Machine Learning (ML) algorithm to automatically identify AF events. The proposed approach, named MORPHYTHM, has been validated on the Physionet AFDB database[1]. The experimental results show that MORPHYTHM improves the classification accuracy of AF episodes, compared to the reference state-of-the-art. The accuracy of MORPHYTHM has been further improved by integrating a local prediction technique. The latter aims at refining the data in order to obtain a more specific training. An empirical evaluation of the new approach, named LOCAL MORPHYTHM, showed significantly better results in the classification process than MORPHYTHM, in particular regarding the increase of True Positive and the reduction of False Negative (critical parameter in medical contexts). From the two previous approaches, however, it was difficult to research and detail the cause that generated a fibrillating episode. In order to realize an Explainable ML approach, a further method has been defined, named AMELIA, which aims to simulate the behavior of a cardiologist, considering two sources of ECG infor-

[1] https://physionet.org/content/afdb/1.0.0/

mation in a combined way: heart beat morphology and heart rhythm. AMELIA is basically composed of two components: one integrating an LSTM Recurrent Neural Network (RNN) and the second integrating a rhythm analyzer. When the RNN reveals a heartbeat with an abnormal morphology, the rhythm analyzer is activated to verify whether there is a simultaneous and contextual arrhythmia. AMELIA has been tested using known state-of-the-art databases, such as the AFDB and NSRDB[2]. The results obtained are ameliorative, especially for some patients and with regard to sensitivity. Beyond AF, in the present project, a real-time approach was defined for the classification of heartbeats into different classes of arrhythmia (including branch blocks, ventricular extrasystoles). The accuracy of the approach was compared with one of the best and most recent works in the literature. The results obtained show that our approach, named NEAPOLIS, provides a more accurate detection of arrhythmia conditions.

It should be emphasized, however, that an electrocardiographic trace is data that must be represented with high accuracy because relevant clinical features may need very high precision to be evaluated. With the huge spread of wearable medical devices, this factor represents a limitation. Indeed, the weight of an ECG signal affects the transmission of wearable devices (e.g., in terms of data rate) and the space occupied in server-side memory, in software systems dedicated to the healthcare. Therefore, approaches for analyzing data on compressed ECGs have been defined. The first approach aims to retrieve vital information from a compressed ECG signal, focusing on the identification of R-peaks as potential key information to estimate the heart rate. Subsequently, an approach, named RENEE, was defined that can classify the morphology of electrocardiographic beats into five classes, as described by the AAMI standard[6], on a compressed ECG through 1-bit quantization.

Software development is an intellectual activity requiring creativity and problem-solving skills, which are influenced by affective states [80]. Even said so, the contexts of motorsport and Software development seem far to each other. Indeed, while in motorsport the subjects have to provide excellent performances in a time-bound scenario, software development is more a day-to-day activity where goals are achieved piece by piece and what matters is the sum of the parts,

[2]https://physionet.org/content/nsrdb/1.0.0/

and not an extraordinary performance of one session. In addition, motorsport requires very fast reaction time and decision. The common factors between these two contexts can be the capability of keeping a constant level of concentration and of keeping a mind free of negative thoughts that can distract the driver or the developer. In the context of software engineering however, there is a situation which has more commonalities with motorsport and that is the management of outages. In these situations, engineers have to act under extreme stress and time pressure, taking hard decisions fast because (depending on the specifics of the context) every extra second of downtime/outage can have very high cost.

Therefore, software development can be intended as an activity that requires a high psychophysical effort, in terms of day-by-day improvements or—in some cases—of production under stress.

Based on the above parallelism, we have defined the following research question:

> RQ_4: *Is it possible to predict the outcome (buggy or not buggy) of a coding task using biomedical and code-based features?*

With the experience gained in different fields of study, with this RQ we aimed at experimenting the biometric monitoring in the context of software engineering. More specifically, while programmers are involved in coding tasks. The promising results obtained in motorsport have suggested the definition of an approach that, through the analysis of biomedical data, is able to evaluate the performance of a developer during coding activities. The performance of a developer is measured as the ability of producing source code that passes all the test cases designed for it. This can be seen as the capability of a driver to conduct a perfect race. The goal, then, is to predict, even before committing the changes and then concluding the programming task, whether the produced code is buggy or not. To this end, a study was conducted involving 20 computer science students and programmers. Each subject was subjected to different programming tasks, divided by type (bug fixing and implementation of a new feature), by difficulty and by time of day in which he performed it (morning or afternoon). Each subject wore a single frontal electrode brain sensor and a wrist sensor (for electrodermal and heart rate measurements) and every activity performed on the desktop was recorded, via mouse

and keyboard tracking. In addition, subjects had to answer some questions after the tasks. These sources of information yielded a dataset composed of numerous features, including features based on (i) heart rate, (ii) skin electrical activity, (iii) brain attention, (iv) keyboard activity, and (v) mouse activity.Preliminary results show that the collected data allows to discriminate with fair accuracy the outcome of a programming task, reaching an accuracy higher than 80%.

1.1 Contributions

The main contributions of this work are:

- design and definition of a set of metrics dedicated to provide an objective measure of a mental performance.
- implementation of a software tool to support motorsport trainers and drivers during simulated races.
- design and definition of several automatic detectors of pathological cardiac conditions to be intended as support to specialized medical staff.
- design of a study that involved computer science students and professional developer with the aim at implementing a tool able at predicting the correctness of a coding task, given monitoring data.

1.2 Structure of the thesis

The present thesis is composed of four main parts: part I describes the background and related works for each of the fields of study described above. Then, part II is dedicated to the description about how we defined a biometry and a Trainer Support Tool based on the integration between body and vehicle data. The final content of this part of the thesis is dedicated to the monitoring of drivers' health state through the automatic analysis of digital ECG. In continuation, part III reports on the work done in the context of the monitoring of software developers. Finally, part IV concludes the thesis.

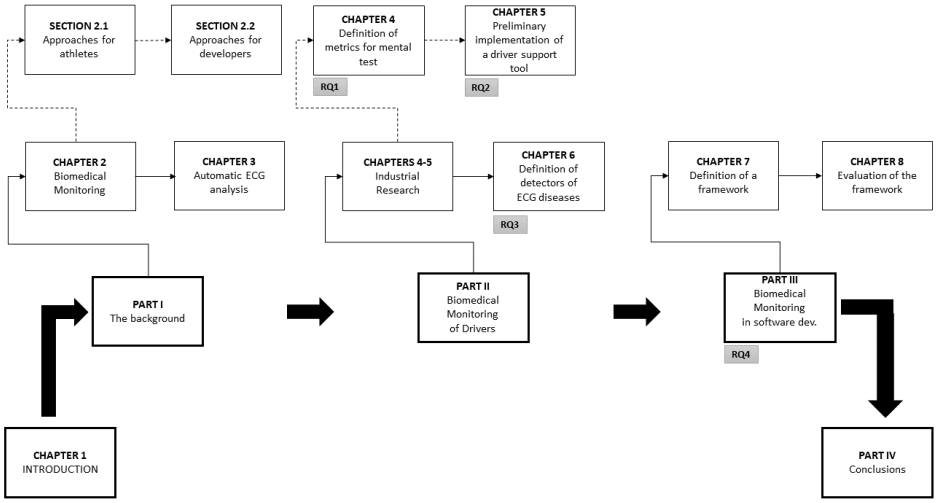

Figure 1.1: A road-map of the thesis.

Conceptually, the thesis can be also divided in the following manner:

- Part I (chapters 2-3): the background of the whole research is proposed;
- Part II (chapters 4-5): the contribution in the industrial research is described;
- Part II (chapter 6): the works conducted on the automatic analysis of an ECG trace are reported;
- Part III: the monitoring in the software development context.

However, to support the reader among the content of this thesis, a road-map is offered in Figure 1.1.

Part I

Background

CHAPTER 2

Biomedical Monitoring

Contents

2.1 Approaches for Athletes

Physical and physiological features are measured and statistically evaluated by biometrics. In sports, much focus is given to the assessment and monitoring of physical and physiological characteristics in order to determine performance and recovery of athletes. Indeed, many technologies are dedicated to the identification of such information. And many of these technologies are packed for storage and analysis in the cloud or on a computer, tablet, or cell phone on wearable devices that upload data (via Bluetooth or other wireless technology). Some systems

rely on wearable global positioning system (GPS) trackers that track acceleration, deceleration, directional changes and jumping (both height and frequency measurements) in real-time during play (e.g., Catapult Sports). Catapult Sports claims to measure athlete's "risk," "readiness," and "return to play," calling itself the "Google Analytics for sports," and counts at least half of the NBA and NFL teams as clients[1].

Other systems track physiological markers and behaviors in different formats. The Zephyr Bioharness, for example, is worn around the torso and tracks heart rate and heart-rate variability (HRV), breathing rate, and movement. WHOOP, a wristband device, tracks similar biovariables and is marketed as a "performance optimization system" that gives scores for strain, recovery, and "sleep performance," as well as to predict performance[2]. Some devices use adhesive bandage-like patches with sensors (e.g., WiSP: Wireless identification and sensing platform) that continuously measure variables such as heart rate, respiration, motion, blood oxygenation, brain activity, muscle function, body temperature, and changes in blood pressure; some track a "whole library" of chemicals present in sweat, including electrolytes, proteins, and heavy metals [76]. Inventors imagine "a pathology lab on your hand" (as quoted in Swetlitz[3]). The non-invasive, compact biomonitoring capability to continuously gather data over long periods of time is the most significant groundbreaking feature of these technologies [118].

While many technologies and devices are widely available, they typically only support classification and measurements for stroke/swing, or they only support metrics for fitness monitoring, such as steps, calories, heart rate, etc. The wearable tracker facilitates tennis stroke classification and simultaneous assessment of heart rate and temperature. There has been a lot of work performed previously using a visual method in the area of tennis stroke recognition [242, 47]. The ITF (International Tennis Federation) allowed the use of a video system with multiple calibrated high-speed video cameras, stationed around the court, and computer software for video analysis for Grand Slams and other major tennis competitions. The method is very costly and can only be used at a time [175] on one court.

[1] Westover, B. 2020. Catapultsports.com
[2] WHOOP. 2020. http://whoop.com/
[3] https://www.statnews.com/2016/01/27/sweat-wearable-tech/

Other approaches and values, such as using an inertial measurement unit (IMU) for tennis match post-analysis applications, are more suitable for daily use.

There are also some smart wearable applications that use four distinct principles of integration: (i) the sensor is incorporated into a tennis racquet handle; (ii) the sensor is connected to tennis racquet strings (such as a string vibration dampener); (iii) the sensor is attached to the tennis racquet grip and (iv) the sensor is attached to the wrist of a tennis player [128].

The scheme of Hallberg *et al.* [94], instead is designed for skiing. The device was designed to track cross-country skiers' heart rate and position/altitude information and uses Bluetooth to relay data to the athlete's Bluetooth/general packet radio service (GPRS-enabled cell phone enhancement)-enabled mobile phone. It is then possible to post the data transmitted by GPRS to a dynamic website so that the success of the skiers can be tracked through the internet. Evaluating load, healing, and the severity of injury is another common application of biometric technology in sport. These evaluations have traditionally relied on self-reporting of pain and effort, necessarily arbitrary metrics, and those that athletes can deliberately exploit to preserve play time or stand on the team, or, among other factors, look stoic.

The rise in the proliferation of ostensibly "objective" interventions poses other questions, especially if they are not properly understood. After a professional football club has recently implemented the system for use in competition, one sports scientist relayed how the abuse of biometric data led to two players suffering season-end injuries in the preseason. The force coach and assistant strength coach had been instructed on their use by the scientists and a business official. The head physical trainer, who had not been educated to use the technology, took care of its introduction at the outset of the preseason. This involved calculating a subjective "load score" which is a measure of the "work" of the general participant. Athletes in the sport have varying loads depending on both their individual efficiency of movement and their position's movement demands. The head trainer wanted to quantify players participating in high-intensity efforts and "average" player load, which he thought would be a reasonable benchmark for the entire team. If any player hit 85 % of that load, the player was excluded from practice by the manager. Players generate loads at very different speeds due to

human heterogeneity. Those collecting tons more easily were put out of practice irrespective of whether they felt exhausted. Those that more steadily amassed loads were kept in practice. The players who were kept in practice the longest were under even more physical burden than those who were taken out, partially because the workloads in the positions of the excluded players had to be assumed now as well. Eventually, by using technologies that were meant to shield them from injury, the only two players left in the position suffered serious hamstring injuries. The teacher refused to realize that the original load baselines of the athletes vary, that collective averages should not be extended to individual athletes, and that linear quantification and regression analysis should not be used to deter harm in a dynamic human environment.

2.1.1 Monitoring in Professional Motorsport

High-speed driving can be viewed as a physiological condition in which motion and motor input must be interpreted in a much quicker and more challenging manner by the brain. Interestingly, recent research gathered in numerous highly qualified groups, including competitive athletes, shows that the skills subtending extraordinary driving abilities, such as those exhibited by experienced Formula racing-car drivers, could be correlated with unique improvements in the brain's morphological and functional architecture [237, 207, 34]. Driving is thus a dynamic activity that needs several cognitive functions to be combined [19]. A scientific study was undertaken by Bernardi *et al.* [19] to understand the morphological and functional brain architecture in skilled competitive driving. Here, both structural and functional brain magnetic resonance imaging (MRI) scans were carried out by 11 professional racing-car drivers and 11 "naïve" volunteers. Short videos showing a Formula One car competing on four different official circuits were introduced to the subjects. Brain activation was analyzed using an Inter-Subject Correlation (ISC) approach in terms of regional reaction and regional interactions through functional connectivity. In addition, particular structural variations between the two groups and possible associations with functional differences found by the ISC study were defined by voxel-based morphometry (VBM). Trained drivers demonstrated a more consistent recruitment of dedicated regions for motor regulation and spatial orientation, including premo-

tor/motor cortex, striatum, anterior and posterior cingulate cortex and retrosplenial cortex, precuneus, middle temporal cortex, and parahippocampus, compared to non-experienced drivers. In comparison, some of these brain areas, including the retrosplenial cortex, had an elevated density of gray matter in experienced car drivers as well. In addition, the retrosplenial cortex, historically correlated with observer-independent spatial map storage, showed a clear association with the performance of the individual driver in official competitions. These results show that in highly qualified racing-car drivers, the brain functional and structural organization varies from that of subjects with normal driving experience, indicating that unique anatomical-functional changes can hinder the achievement of outstanding driving performance [19].

2.2 Approaches for Developers

There are several ways to quantify the cognitive and emotional mechanisms that are involved when software is being created by humans. Any of these offers indirect ways to obtain further understanding, such as happiness [85] self-report assessments or quantitative results on cognitively difficult activities. Current developments in biometric sensor technologies (aka. psycho-physiological) provide new possibilities to gather and analyze a wide range of clear, accurate data about a human being and her cognitive and emotional states when functioning and creating applications. The fundamental theory is that the psychological states of a person are related to their physiological processes and that biometric sensors can be used to assess these physiological processes. Any biometric measurements, including skin-, pulse-, eye-, and brain-related indicators, have already been investigated and associated with the cognitive and emotional states of a person through studies in psychology and other fields. For instance, researchers have found that pupil size and electrodermal activity (EDA) can be linked to cognitive load [210, 92, 108].

Software engineering science is now beginning to take advantage of biometric measurements using a range of sensor tools to analyze software developers' cognitive processes. These experiments vary from the usage of an eye tracker to catch the eye fixations of a creator while reading or navigating code [211, 121], EDA

wristbands, EEG monitors, or chest belts to capture skin conductivity as well as brain- and heart-related measurements to measure mental load[171, 71, 73], all the way to the use of functional magnetic resonance imaging (fMRI) and near-infrared spectroscopy (NIRS) to investigate brain activity patterns during software comprehension [214, 104],

As biometric sensor technology becomes less invasive, easier to incorporate into the work of the developer, cheaper, and more precise, in real-time and in real life settings, we are now able to collect more fine-grained biometric data from app developers. These developments would allow us not only to better understand a developer when operating, but also to improve and provide developers with more instant support. For example, we could minimize interruptions from other co-workers at inopportune moments by observing when a developer feels a heavy cognitive load, or be able to interfere until a developer develops a flaw. In addition, we will be able to achieve a simpler view of the development of applications that could lead to developments we cannot foresee today.

Overall, the findings of studies on the use of biometric measures in software engineering have already demonstrated the promise of such data. Around the same time, before the use of biometric data can become generally embraced, there are still a number of issues to overcome. These problems range from the precise analysis of the data, its noisiness, sensor restrictions and invasiveness, to developers' privacy concerns. Low heart rate variability, for example, may normally be related to a person's stress and high cognitive load, although it is not possible to distinguish whether this is due to stress, a high cognitive load, or both when a low heart rate variability is observed. The drawback and noisiness of current low-invasive heart rate sensors that use optical sensing is another example. Since gestures or changes in ambient illumination impact an optical sensor, it is difficult to collect heart rate variability data correctly and consistently for such sensors, especially when they are integrated into wristbands and users move around a lot [65].

Skin temperature and electrodermal activity are widely used measures related to skin (EDA). In the next phase, we will concentrate on EDA. Electrodermal activity (EDA) is a human skin property that allows its electrical characteristics to differ continuously. More precisely, the skin's ability to conduct electricity

differs and is related to a person's level of psychological or physiological arousal. It is possible to infer, e.g., the degree of stress encountered by an individual, by calculating electrical properties such as resistance. It provides a degree of clear insight into the autonomous management of emotions since EDA is not under voluntary supervision. Measuring EDA at different body sites, however, produces distinct outcomes, and EDA responses are delayed 1-3 s, meaning that assessing mental behavior from an EDA signal is not straightforward. Generally, the EDA signal can be divided into two parts: the low frequency, tonic part, slowly evolving, and the quickly adjusting, high frequency, phasic part [202]. The mean value is one of the widely used metrics for the tonic part, while the commonly used characteristics for the phasic part are connected to the signal peaks.

It is possible to quantify brain function in different ways, though the measures that can be captured rely heavily on the instrument that is being used. These instruments differ in the nature, precision, and granularity of the information they obtain, but also in their invasiveness, reducing the types of studies that can be conducted. A tool for monitoring electrical activity in the brain is electroencephalography (EEG). Electrodes mounted on a participant's scalp test variations in voltage that indicate neuronal activity. If the electrodes are positioned appropriately, it is possible to interpret the various signals from the several electrodes to provide activation information in different brain regions. In order to investigate neural mechanisms, this knowledge can be used.

Brain wave frequency bands called alpha (α), beta (β), gamma (γ), delta (δ), and theta (θ) are widely used measurements extracted from an EEG. Each of these brain wave frequency bands has a particular frequency spectrum and intensity, and under varying conditions, they show more or less behavior. For example, when a person is in a calm state, alpha waves may usually be detected, but the alpha waves either vanish or dramatically decrease their intensity as soon as the physical or mental activity increases [225]. EEG has a poor spatial resolution, which ensures that only rough information on activations in various brain regions can be given. However, it is less invasive than other sensor systems, has high temporal resolution, and can be very flexible and thus used in cases when participants are traveling.

Commonly used heart-related measures are the heart rhythm (HR), the heart rate variability (HRV), and the blood flow pulse (BVP). The heart rhythm corresponds to the number of heart contractions per minute and the amplitude of the heart rate reflects the difference of two successive heart beats in the time period. The blood volume pulse tracks the movement of blood into various areas of the body and can change, for example, due to stress [225], as the sympathetic nervous system increases its operation. The mean pulse rhythm, the mean and the standard deviation of the time between two heart beats, and properties that capture the BVP signal peaks are typical aspects of these metrics. Today, there are different methods to collect measurements related to the heart that again vary in the form, granularity, and precision of the collected data. Tools range from widely used wrist watches, such as Apple Watch[4], to chest braces or arm bands, all the way to electrocardiograms that use specialized electrodes to capture heart pressure using an optical sensor. The accuracy of the measurements is especially influenced by the sort of sensor used and the position of the sensor(s). For example, it is difficult at best to collect correct HRV data with a wrist watch using an optical sensor: firstly, the wrist is normally used and shifted a lot and secondly, wrist watches are also not securely attached to the wrist so that the optical sensor moves about a lot.

Before using biometrics to research or help app developers, there are many points to be considered. The choice of biometric sensor or physiological measurement, the human variations in the measurements, the analysis of data, and how to better use them for a sample or in the field [65] are some of the most prominent of these points.

In a study on short code comprehension and error localization operations, the researchers used a combination of fNIRS and eye tracking and observed that linguistic antipatterns greatly enhanced the cognitive load of developers [67] in the source code.

In a longer-term study conducted over the course of a week in conjunction with computer interaction data, a combination of heart, breathing, and skin-related measurements from two different sensors was obtained to predict the sophistication of code elements and the accuracy of the produced code. Using a

[4]https://www.apple.com/watch/

machine learning method, the various biometric readings were used to construct a prediction model that could only surpass a model based on traditional (non-biometric) measurements [73]. Study has also looked at the use of biometric sensors to analyze developers' emotions. The use of multiple sensors is capable of benefit from these experiments, as emotions manifest in many physiological responses.

A study that integrated non-invasive, low-cost EEG, EMG, and GSR (EDA) sensors found that correct classification of emotional valence and arousal was possible using machine learning classifiers on the sensor data [79]. A research using a series of biosensor data (EEG, EDA, skin temperature, heart rate, blood volume pulse, eye tracking) to create a machine learning model showed that during program transformation tasks [167], it was possible to distinguish between positive and negative emotions and low and high growth. In general, though, arousal prediction is stronger than valence prediction——the positive or negative character of an emotion—-the volume of stimulation associated with an emotion, which has already been seen by several studies in other fields. Finally, studies have often explored variables, such as interruptibility, seen as a developer's availability of an interruption[253]. For example, one study explored how biometric and interaction data could be used in an office environment to predict interruptibility (Züger *et al.* 2018). Computer interaction data was found to be more reliable than biometric data alone, but provided the best results for a mixture of both[65].

The software engineering research community has involved the use of physiological signals to investigate the relationship between developers' cognitive and affective states and several aspects of software development, like code comprehension [183], productivity [191], and software quality [165].

Fritz *et al.* [72] combined different biometric features for predicting the difficulty of a task. Using eye-tracking, EDA and EEG, the authors can predict whether a developer perceives a task as difficult task both *post hoc* and as the developer is working. Their experiments show that it is possible to predict whether a new developer will be experiencing difficulties in a code comprehension task with a precision of over 70% and a recall over 62%. Parnin [182] studied the complexity of programming tasks by relying on the analysis of sub-vocal. He used an electromyogram (EMG) to show how those signals correlates to cognitive

patterns involved in dealing with a programming task. Fucci *et al.* [74] use EEG, EDA, and heart-related measurements for the automatic identification of code comprehension tasks. They developed an approach based on supervised machine learning to automatically identify what kind of tasks developers are working on. Their findings show that lightweight biometric sensors can be used to accurately distinguish between code and natural language comprehension tasks.

Recent research also investigated the relationship between physiological measures and developers' productivity. Radevski *et al.* [191] proposed a framework for continuous monitoring of developers' productivity based on brain electrical activities. Müller and Fritz [165] used a combination of different biometric measurements to predict self-assessed progress and interruptibility of developers while programming. They also demonstrated that it is possible to classify developers' emotions using biometrics a combination of EEG-based, eye-related, and heart-related metrics with an accuracy of 71%. The progress experienced by developers was predicted at a similar rate, but using a different set of biometrics (i.e., EDA signal, skin temperature, brainwave frequency, and the pupil size). The findings of the study by Müller and Fritz [165] have been confirmed and extended by a recent replication of their study performed by Girardi *et al.* [81]. They investigated the range and triggers of emotions that software developers experience while programming and confirm the positive correlation between emotional valence — i.e. the (un)pleasantness of the emotion stimuli — and the self-reported progress, previously reported in previous independent studies [165]. Furthermore, they demonstrate how emotion recognition is feasible also using a reduced set of biometrics including EDA and heart-related only, collected using a single device, *i.e.,* the Empatica E4 wristband. They trained a machine learning classifier that can detect valence with an accuracy of 71% and arousal — i.e. the emotional activation, ranging from excited to relaxed — with an accuracy of 68%.

Müller and Fritz [166] also investigated the use of EDA, EEG, and heart-related biometrics for real-time identification of code quality concerns in a real-world setting. They the authors identified difficult parts of the system—e.g., low-quality code containing bugs. The authors provide some evidence that biometrics can outperform traditional code-related metrics to identify quality issues in a code base.

As far as code comprehension is concerned, two similar studies, by Siegmund *et al.* [215] and Ikutani and Uwano [105], assessed the brain activity of developers involved in code comprehension tasks. Siegmund *et al.* [215] used fMRI to show clear activation patterns in five regions of the brain all related to language processing, working memory, and attention. The study by Ikutani and Uwano [105] uses near-infrared spectroscopy to show that different parts of the brain are activated during code comprehension with respect to a specific sub-task. For example, they distinguish between the areas activated by the workload necessary to memorize a variable and the ones activated by arithmetic calculation. More recently, Peitek *et al.* [183] used fMRI to monitor the brain activity of 28 participants involved in the comprehension of 12 source code snippets. Their results show that distinct areas of the brain are activated during such a task. Moreover, the activation patterns suggest that natural language processing is essential for code comprehension. To get a more comprehensive view of the strategies adopted by developers when comprehending source code, Peitek *et al.* [183] obtained simultaneous measurements of fMRI and eye-tracking devices. They showed strong activation of specific brain areas when code beacons are available. However, their setup was subject to data loss—complete fMRI and eye-tracking data could be collected for 10 out of the 22 participants.

Also Lee *et al.* [146] used biometric devices with the goal of predicting task difficulty as well as developers' expertise. Analogously to the aforementioned studies, they showed to participants several snippets of code (23 in total) asking to reflect and think about their output. 38 between expert and novice programmers were involved in the study and their brain activity was captured using a 16 channel EEG amplifier, called V-amp. In addition, they used a SMI RED-mx eye tracker to collect information related to the eye-gaze. At the end of the experiment, participants answered a questionnaire reporting the own perception of task difficulty for each task as well as their level of experience with i) general programming, ii) the use of different languages and iii) the estimation of experience in comparison with classmates. The authors, then, used the self-assessment ratings to categorized both the perceived task difficulty and the developers' experience in two labels, that are a) easy and difficult for the perceived task difficulty and b) expert and novice for the expertise. As such, they trained two machine learn-

ing classifiers using the support vector machine and performed the 10-cross-fold validation. Both for task difficulty and expertise the classifiers achieved the best performance in the setting including the combination of EEG and eye tracker features. Specifically, the average accuracy between the 10 folds were 68.8% and 96.4%.

A more recent work based on the idea of monitoring the cognitive load and the mental effort to support developers is the study by Couceiro *et al.* [49]. Specifically, the authors investigated the use of HRV and a pupillography together with the eyetracking to identify which lines of code result more difficult to comprehend and, consequently, require higher mental effort. In line with the work by Mueller and Fritz [168], the authors considered the use of biometrics as a way to reduce the possibility of introducing bugs, suggesting which lines of code require a second look before the commitment. During the experiment, 30 programmers had to deal with 3 different Java programs, having different complexity levels associated. The set of sensors used in the experiment consisted in the Biosignalplus toolkit, through which the HRV has been collected and SMI Senso Motoric Instruments eye tracker. The authors analyzed changes in the HRV and in the pupil diameter since they can be considered a symptom of a mental effort and identified critical regions of code.Then, they compared the critical regions found with critical regions manually annotated by experts. From this analysis, they conclude that the eye tracking could be used to pin point lines of code which require high mental effort and that could be buggy.

Also the effect of mood on the debug activity was tested in the work proposed by Khan *et al.* [122]. The authors designed two experiments. In the first, programmers were shown short movie clips that were chosen for their ability to evoke unique emotions. After that, they ran a debugging exercise. The video clips had a major impact on programmers' debugging results, with a significant difference after watching low-arousalevoking and high-arousalevoking video clips, according to the findings. The mood of programmers was influenced in the second experiment by telling participants to dry run algorithms for at least 16 minutes. They did some physical workouts before returning to the dry running algorithms. After the physical workouts, the findings revealed a substantial rise in arousal and valence, as well as a boost in programmers' job performance. This indicates

that programmers' moods have an effect on such programming activities, such as debugging.

It was also investigated how Affective Events Theory (AET) can impact the processes in the IT world. In the study proposed by Shaw *et al.* [212], the scientific basis is the AET. This theory says that both affect and judgment-based processing may have an effect on job efficiency. Although perceptions play a role in this process, emotions play a direct role in certain outcomes. Moreover, feelings are reactions to job activities that workers witness on a regular basis. Affective Events Theory, according to this work-in-progress, may be a valuable complement to the nomological network of IT Human Resource science.

In the work proposed by Wrobel *et al.* [238], it was conducted a study on software developers' emotional interactions at work. A significant number of software developers who were surveyed were entitled to have fair responses to study questions. The majority of respondents said anger was the most prevalent negative emotional state they encountered. This emotional state was also ranked as the most bothersome in high productivity, making it the most dangerous. Rage was another mental state to which further studies will be paying careful attention. It was the only negative condition that a large portion of the respondents rated as increasing competitiveness. Developers described an excited mood as one of the positive emotions. It was determined to have the biggest beneficial effect on productivity. Around the same time, it was a regular phenomenon in respondents' workplaces.

2.2.1 Our study and the literature

As described above, biometric data has already demonstrated its validity in the field of Software Engineering. The aim is to find out the different ways to quantify the cognitive and emotional mechanisms that are involved when software is being created by humans. Most of the effort from the scientific literature has been dedicated to the biometric monitoring of developers during task of code comprehension in order to assess the factors that are involved in this specific type of software engineering activity. Also, the scientific literature has included, in its efforts, the investigation on the interaction between developers' cognitive

and affective states and different facets of software creation using physiological signals.

With respect to state-of-art works, our approach—presented in chapters 7, 8—uses biometric data to identify the correctness of a coding task. To the best of our knowledge, this is the first work that tries to assess a measure of performance of a developer by taking into account several biometric measurements.

CHAPTER 3

Automatic ECG Analysis

Contents

3.1 Introduction

Fatigue is a type of pain that human beings experience. It is a sensation that occurs after a lot of physical or emotional activity is done by the human body [48]. It would decrease the willingness of individuals to complete work. Generally speaking, exhaustion is categorized into emotional fatigue, physical fatigue, and pathological fatigue. Exercise-type exhaustion is described in the Fifth International Conference on Sports Biochemistry, held in 1983, as the body is unable

to sustain a function at a certain degree, or cannot maintain a predetermined strength of exercise. Many scholars around the world have embraced this definition and it is commonly followed. The heart load of the person can begin to rise and become more irritable than normal as sport exhaustion happens, body becomes soreness and lacks endurance, thought slows down, decision and response become sluggish [62].

Sport fatigue is also a comprehensive physical and mental fatigue that incorporates physical, mental, and psychological strength [40]. The heart is the human body's most important organ. The cardiomyocytes frequently depolarize and repolarize to form a galvanic couple with the neighboring cell membrane under the cycle of the equilibrium of anion and cation concentration in the cell membrane and deliver daily current pulses in the heart. This pulse can activate the muscle cells in the ventricles and atria, causing the heart to contract and relax regularly, make the heart a blood pump for the human body, and allow blood to flow throughout the body with adequate strength to sustain the human body's everyday intake.

The depolarization and repolarization mechanism of each cardiomyocyte can be considered as a dipole field in each cardiac cycle, and the human body has conductivity and can be regarded as a conductor of volume. In the human body, an electric field is produced, which causes a potential difference in electrocardiograph (ECG) signal production. An significant bioelectric signal is the electrocardiograph (ECG) signal, closely linked to sports fatigue. It is ideal for evaluating sports exhaustion and researching it. By way of a dedicated electrode and amplification [55, 150], an electrocardiograph (ECG) signal can be obtained on the surface of the human body.

In many nations, the development of medical technologies and the gradual rise in average life expectancy contribute to the obvious movement towards an ageing population [228]. The increase in health spending is caused by an ageing population. The stresses on health and elderly care services are also quickly germinating [129]. The report indicates that about 75% of the elderly have one or more chronic illnesses, one of which is one of the most prevalent diseases of cardiovascular disease. Cardiac patients usually rely on routine health checks to determine their status. But the testing time in the hospital is short, and the

possible risk of the illness will not be diagnosed. Early alarm and avoidance can only be triggered by long-term, constant tracking and the recording of frequent cardiac physiological signals.

In the United States, more than 28% of elderly people (8.8 million women, 3.8 million men) actually live alone [172]. Elderly health management is therefore becoming a serious concern. In order to satisfy this need, several popular healthcare apps have been implemented to track the electrocardiography of users in recent decades. However, there are also certain disadvantages to these systems [244]. For example, the Holter, a handheld instrument for 24-72 hours of continuous tracking of the heart's electrical operations, may provide non-real-time records of 12 ECG leads. But for long-term use, the procedures are awkward and unpleasant, so it's not acceptable for elderly home treatment. The holter can only be used in clinical diagnosis because of the constraint of electrode design, size and record price, but lose the end-user perspective [163, 197].

Methods based on machine learning, especially neural networks, are the most frequently used methods for solving clinical problems in the field of medicine [119]. Often, they include diagnosis and monitoring for prognosis. Therefore, these machine learning methodologies can be used to assimilate multiple patterns and can become the knowledge base for computer-assisted decision support systems (CDSS). While it is challenging to develop such efficient knowledge-based clinical decision support structures, at the time of decision-making, their generation can provide recommendations and can be part of the usual clinical workflow [119].

Thus, the use of clinical decision support systems (CDSS) has been proposed in order to support decision-making, to prevent medical mistakes and to increase patient safety. The efficiency of six distinct CDSS systems [4] is identified and contrasted in a past study on clinical decision support systems for heart diseases. Another research explains the tools used for cardiac attack prediction and their nuances [37, 201].

In this PhD thesis, the focus is given to some specific heart pathological conditions, such as Atrial Fibrillation, Premature Ventricular Contractions (PVC), Bundle Branch Block (BBB), Atrial Premature Beat (APB) and other types of ECG morphology characterization according to a specific standard. The next sec-

tions are intended to report on the literature background on the analysis related to the focus of this thesis.

ECG extraction and segmentation of features play a crucial part in the detection of most cardiac diseases. The key aim of this part of the study is to report the different approaches based on machine learning to diagnosis interesting pathologies [197].

3.2 Atrial Fibrillation

Atrial Fibrillation (AF) is a quite common yet dangerous cardiac pathological condition. The numbers say that in the UK, almost 534k people have contracted this disease, in 1995 [218]. In 2010, the estimated numbers of men and women who were affected by AF world-wide were respectively 20.9 and 12.6 million. Moreover, the incidence was higher in developed countries, such as Europe and US. Indeed, it is expected that - by 2030 - the number of AF patients will be between 14 and 17 million only in Europe [124]. Besides, such a condition is very expensive: the direct cost of healthcare for patients affected by AF was about ∼655M in 2000, equivalent to 0.97% of the total UK National Health System (NHS) expenditure [218]. While in US, it has been estimated that the medical cost caused by AF is $26 billion annually [110]. Also, the prevalence of the disease is expected to more than double in the next 50 years as the population grows older [156].

Most of the cost of healthcare for patients affected by AF is due to hospitalizations and home nursing. In this context, telemedicine would be very helpful. Indeed, telemedicine would allow to remotely and continuously monitoring thousands of patients. However, telemedicine alone is not enough: remote monitoring could help reducing the global cost, but physicians and nurses would be still required to perform such a task.

The best way for reducing the cost of AF for NHS through telemedicine would be by employing automated approaches for AF detection: a software system constantly acquires data from the patient and, when an anomalous condition is detected, physicians are warned [11]. This would allow to reduce the number of specialized personnel that is required to monitor the patients.

AF is a pathological heart rhythm which results in a rapid and irregular beating of the atria. The consequences of this cardiac disease are very adverse. Indeed, contracting AF may lead to stroke, dementia, and death. Thus, a precise diagnosis of this pathology needs to become a priority [203].

During AF, the hearts atria are quicker than normal beating. This leads to the condition that the blood is not ejected completely out of atria and there might be chances of formation of blood clots in the atria. The result is an increased risk of stroke. Electrocardiograms (ECGs) are useful tools for AF detection. ECGs are recordings of heart's electrical activity and are widely used by physicians to diagnose pathologies related to the heart. Patients with or at risk of cardiovascular diseases often present ECGs that are irregular in rate and in morphology of the signal [38].

In the last decade, several methods have been proposed for the automatic detection of AF. Most of them have shown good results by exploiting only the analysis of heartbeat rhythm [245, 208, 46, 158, 239]. Morphological features were used in the patent by [131] and, even not specifically focused only on AF detection, in the work by [241].

For sake of space limitation, in the following we focus the attention on the most accurate methods reported in the literature, *i.e.,* the ones summarized in Table 3.1. These methods represent our baseline, due to the common evaluation on the Physionet MIT-BIH AF Database [82]. This database includes 25 long-term ECG recordings of patients with atrial fibrillation (mostly paroxysmal[1]). Of these 25 long-term ECG recordings, 23 include the ECG signals while for records (*i.e.,* patients) 00735 and 03665 only information on the rhythm are available. The individual recordings are 10 hours each in duration and contain two ECG signals each sampled at 250 samples per second with 12-bit resolution over a range of $\pm$ 10 millivolts.

Huang *et al.* [103] propose a method to detect the transition between AF and sinus rhythm, based on RRI. In the proposed method the authors first obtain

[1]AF can be classified into specific types depending on the duration and ability to self-terminate or to be terminated by some therapeutic technique [124]. AF is named as paroxysmal when it is self-terminating (in most cases within 48 hours). Some AF paroxysmal episodes may continue up to 7 days. Thus, also AF episodes that are cardioverted within 7 days are considered paroxysmal.

Table 3.1: Literature Detector Performances on MIT-BIH AFDB. In $AFDB_1$ records "00735" and "03665" excluded, while in $AFDB_2$ records "04936" and "05091" excluded.

Method	Year	DB	Se[%]	Sp[%]
Zhou (2015) *et al.* [249]	2015	AFDB	97.4	98.4
Petrenas *et al.* [184]	2015	AFDB	97.1	98.3
Asgari *et al.* [7]	2015	$AFDB_2$	97.0	97.1
Zhou (2014) *et al.* [248]	2014	AFDB	96.9	98.3
Lee *et al.* [143]	2013	$AFDB_1$	98.2	97.7
Huang *et al.* [103]	2011	AFDB	96.1	98.1

the delta RR interval distribution difference curve from the density histogram of delta RRI, and then detect its peaks, which represent the AF events. Once an AF event was detected, four successive steps have been used to classify its type.

Lee *et al.* [143] introduce a method for automatic detection of AF using time-varying coherence functions (TVCF). The TVCF is estimated by the multiplication of two time-varying transfer functions (TVTFs). The first TVTF is obtained by considering two adjacent data segments (as input and output signals); the second TVTF is computed by reversing these signals. They found that the resultant TVCF between two adjacent normal sinus rhythm segments shows high coherence values (near 1) while lower than 1 if either or both segments partially or fully contain AF, throughout the entire frequency range. They have also combined TVCF with Shannon entropy. In this case, the approach shows even more accurate AF detection rate: 97.9% for the MIT-BIH AF database (considering 23 records) with 128 beat segments.

Zhou (2014) *et al.* [248] devise a method for real-time, automated detection of AF episodes in ECGs. This method utilizes RR intervals, and it involves several basic operations of nonlinear/linear integer filters, symbolic dynamics and the calculation of Shannon entropy.

Asgari *et al.* [7] employ a stationary wavelet transform and a support vector machine to detect AF episodes. The proposed method eliminates the need for P-peak or R-Peak detection (a pre-processing step required by many existing

algorithms), and hence its performance (sensitivity, specificity) does not depend on the performance of beat detection.

Petrenas[184] propose a RR-based AF detector with a low complexity structure. The detector involves blocks for pre-processing, bigeminal suppression, characterization of RR irregularity, signal fusion and threshold detection.

The method proposed by [249] will be deeply explained in the next sections for two main reasons: (i) the method represents our baseline in the evaluation of our approaches; (ii) the entropy measure used in [249] has been exploited as feature in some of our approaches.

Billeci *et al.* [23] used a LS-SVM Machine Learning approach to classify an ECG segment in *Normal*, *AF* or *Other Rhythm*. According to *ANOVA*, they have obtained respectively a F equal to 0.94, 0.91, 0.86 and a global F=0.90. The authors have chosen several types of features, by taking into consideration the difference between the preprocessed and the raw signal, the RRI analysis and the ratio between the QRS amplitude complex and the beat *prematurity*.

Lahdenoja *et al.* [133] have shown a different kind of Atrial Fibrillation detection, based on the use of a smartphone (to be positioned on the chest of the person). They have chosen to deal with two kinds of features, ECG signal based features and RRI based features. A comparison between the performances of different Machine Learning classifiers has been reported. They obtained an overall accuracy of 97.4% in AF vs. healthy classification (a sensitivity of 93.8% and a specificity of 100%).

Here, we discuss in detail the method proposed by Zhou *et al.* [249] because we chose it as baseline for the approaches further reported in the section focused on the detection of Atrial Fibrillation episodes of this thesis. Such a method consists in the following steps.

Step 1: Converting the HR sequence. Considering a preliminary stage of RRI analysis and thus known the HR sequence, the first step expected in the method is to evaluate a symbolic dynamic. This quantity encodes the information of hr_n to a series with fewer symbols, with each symbol aims at representing an

instantaneous state of heartbeating. The mapping function is the following:

$$sy_n = \begin{cases} 63, & \text{if } n \text{ hr} \geq 315 \\ \lfloor hr_n \rfloor, & \text{other cases} \end{cases}$$

where $[\cdot]$ represents a floor operator.

Step 2: Building the symbolic sequence. The authors apply a 3-symbols template to explore the entropic properties of the symbolic series sy_n. Thus, to examine the chaotic behavior, the word value can then be calculated by the operator as defined below:

$$wv_n = (sy_{n-2} \times 2^1 2) + (sy_{n-1} \times 2^6) + sy_n$$

Step 3: Computing the entropy. The authors define a coarser version of Shannon entropy $H^{''}(A)$ to quantitatively calculate the information size of wv_n. In this study, the dynamic A comprises of 127 consecutive word elements from wv_{n-126} to wv_n, as proposed in the function below:

$$H^{''}(A) = -\frac{k}{Nlog_2 N} \sum_{i=1}^{k} p_i log_2 p_i$$

where N and k are total number of the elements and characteristic elements in space A, respectively.

Step 4: Classification. Based on the obtained entropy value, a final beat-to-beat classification (*fibrillant* or *non-fibrillant*) is presented by applying a threshold discrimination. The optimal threshold was empirically identified at 0.639.

These steps are graphically represented in Figure 3.1.

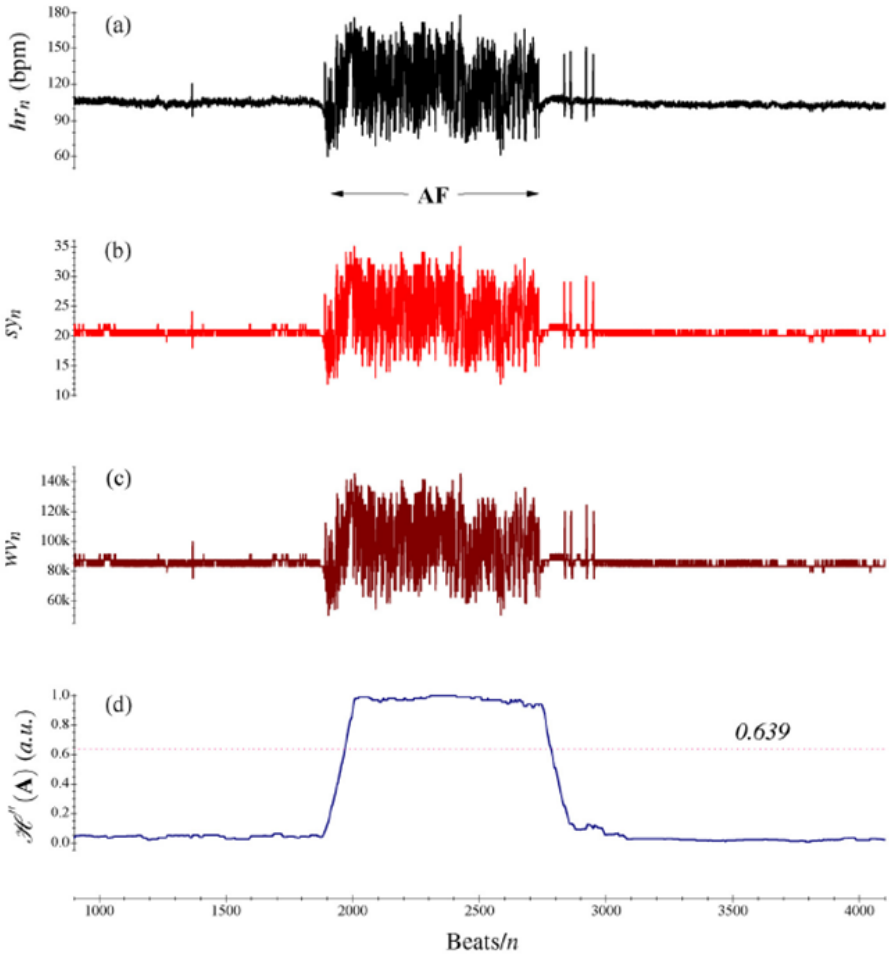

Figure 3.1: Graphical representation of the main steps in the method by [249].

3.3 Arrhythmia Conditions

Arrhythmia can describe a disorder that affects the regularity of the heart rhythm, by observing too fast or too slow rhythm. Arrhythmia can be categorized into two types: atrial and ventricular. Especially this latter kind of arrhythmia may be very dangerous. Therefore, without a continuous monitoring and the right attention, ventricular arrhythmia can lead to sudden cardiac arrest [63].

A lot of effort has been dedicated by the scientific community to the definition of methods for the automatic detection of arrhythmia conditions [115, 223, 216, 9, 181].

Zhao *et al.* [246] proposed an approach for the extraction of features that allows a reliable heart rhythm recognition. They basically used two techniques for the features generation: wavelet was used to extract the coefficients of the transform and autoregressive modelling (AR) to obtain the temporal structures of ECG waveforms. Then, wavelet and AR coefficients were concatenated together to form the feature vector for the classification. They evaluated a large set of outputs that also include our target conditions, but they chose to experiment the method on a subset of the available recordings from the MIT-BIH Arrhythmia[2], a freely accessible and common database of the scientific literature with annotation at heartbeat level. The results showed that the approach provided good performances of classification reaching an accuracy of 99.68%.

Li *et al.* [148] proposed a method for ECG classification using entropy on Wavelet packet decomposition (WPD) and random forests. The authors also experimented the devised method on the MIT-BIH Arrhythmia database but with a different output because they conducted another kind of experiment, focused on a medical standard, *i.e.,* the EC57:1998 standard [6]. The authors stated that although the coefficients by Discrete Wavelet Transform (DWT) or WPD can reveal the local characteristics of an ECG signal, the number of such coefficients is usually so huge that it is hard to use them as features for classification directly. Therefore, they extracted some high-level features from these coefficients for better classification. In the proposed method, they chose the entropy as high

[2]https://archive.physionet.org/physiobank/database/mitdb/

level features extractor from a DWT. The results reported on an obtained overall accuracy approximately equal to 94.5%.

Another very important set of features is the one proposed by Leonarduzzi *et al.* [147], *i.e.,* a set of features derived from the multifractal analysis. The authors stated that this analysis highly suits the analysis of the Heart Rate Variability (HRV) fluctuations, since it gives a description of the singular behavior of a signal. Therefore, the main features of this work are based on the multifractal wavelet leader estimates of the second cumulant of the scaling exponents and the range of Holder exponents, or singularity spectrum. The results demonstrated how these features can be involved in a tool for a precise detection of myocardial ischemia. Many works from the scientific literature have involved the Fast Fourier Transform (FFT) in their methods for the classification of ECG segments. For instance, Haque *et al.* [95] proposed a combination of FFT-based and wavelet features. The main findings achieved by the authors was that the wavelet can provide better indicators—rather than the FFT—of small abnormalities in ECG signals.

Here, we discuss in detail the method proposed by Pandey *et al.* [181] because we chose it as baseline for the approach further described in the section focused on the detection of arrhythmia conditions of this thesis. The choice is not random: the selected approach provides a complete automatic detection of heartbeats in five heartbeat types, including the LBBB, RBBB and PVC, *i.e.,* the same of our proposed approach. The selected method is based on a single Long Short-Term Memory (LSTM) Neural Network as model. The inputs to the model were based on higher-order statistics, wavelets, morphological descriptors, and R–R intervals. Thus, 45 features were in charge of describing the electrocardiogram signals. In details, to extract the features, the authors designed a temporal window of 180 samples sized (half of a second on the MIT-BIH Arrhythmia). The window was centered on each R peak, previously obtained thanks to the annotations of each R wave position available from this database. The features have been evaluated only inside this interval.

3.4 Heartbeat Classification

In the last years, several methods have been proposed for the automatic classification of ECG heartbeats [159, 116, 193, 36, 77, 240, 153]. Most of them have shown good results. They all dealt with the full ECG, *i.e.,* without any type of compression. In addition, most of them involve complex algorithms, *i.e.,* that need high-computational cost for the creation of the features. In details, these approaches classify each heartbeat in five output categories: normal beat (N), ventricular ectopic beat (V), supraventricular ectopic beat (S), fusion of a normal and a ventricular ectopic beat (F) and unknown beat type (Q). Also, they have been validated on the public MIT-BIH arrhythmia database[3].

Mondejar *et al.* [159] proposed a method for the automatic classification of ECG based on the combination of multiple Support Vector Machines. The method relies on the time intervals between consequent beats and their morphology for the ECG characterization. Several features based on wavelets, local binary patterns (LBP), higher order statistics (HOS) were employed. The designed methodology approach was tested classifying four kinds of abnormal and normal beats. The authors achieved an overall accuracy of approximately 0.945 and, in some cases, they have obtained better results than the related state of the art approaches.

Kandala *et al.* [116] presented an inter-patient heartbeat classification algorithm. The foundation of this work is based on the consideration that the ECG is a non-stationary, non-Gaussian signal derived from nonlinear systems [193]. Therefore, the authors employed a decomposition method, namely improved complete ensemble empirical mode decomposition (ICEEMD), to obtain features from the ECG beats. Then, nonlinear measures such as entropies and higher-order statistics (HOS) were determined from the modes obtained after ICEEMD. These were used as features for the discrimination of the heartbeats. To handle with class unbalance, the authors employed a type of ensemble classification based on a majority voting scheme. Finally, the authors demonstrated good improvement — with respect to the state of the art methods — especially for the minority classes.

[3] https://physionet.org/content/mitdb/1.0.0/

Garcia *et al.* [77] proposed a heartbeat representation, called the temporal vector-cardiogram (TVCG), and an optimized feature extraction process with complex networks and particle swarm optimization (PSO). The authors show that their method presents an overall accuracy in the classification equal to 0.924.

Xu *et al.* [240] proposed an end-to-end method with a deep neural network (DNN) for both feature extraction and classification based on aligned heartbeats. We chose to describe here this work because used as baseline in a further reported approach dedicated to the classification of heartbeats in compressed ECG. This selected method avoids any further elaboration for the features creation and produces optimized ECG representation for heartbeat classification. The overall working principle of this approach can be resumed as follows: the system buffers raw single lead ECG signals — at one end — and produces heartbeat classification, at the other end. The pre-processing concerns with the selection of heartbeats from continuous ECG signals. In this approach, a heartbeat signal is represented by a segment of the ECG that comprises the consecutive sample points of a complete heartbeat cycle, which includes not only the QRS complex but also the P and T waves. The Neural Network (NN) is used for both feature extraction and classification, which are achieved by the lower part and the upper part of the network, respectively. Two steps must be performed in order to extract fixed-length feature vectors from raw ECG signals: (i) heartbeat segmentation and (ii) heartbeat alignment. To the best of our knowledge, this method [240] represents one of the best approaches presented in the state of the art with an overall accuracy of 0.947 in the 5-class classification.

3.5 ECG Compression

The widespread use of ECG records, especially as a means of promoting clinical health care from a distance, enhances the importance of dedicated data compression techniques. Without any loss in signal reconstruction, what is referred to as lossless compression, or allowing any distortion that does not change the clinical details of the data, compression of ECG signals can be realized. Lossy compression is called the latter. An ECG signal can be enclosed inside a file con-

siderably smaller than that containing the uncompressed record by this technique [195].

In the literature, several compression methods of ECG signals have been proposed with the aim of reducing the data rate of the IoMT physical device and its energy consumption. The proposed approaches can be classified in hardware-based methods and digital-based methods [186].

The hardware-based compression methods exploit the sparsity of the ECG signal in the time domain to design specific Analog-to-Digital Converter (ADC). According to Picariello *et al.* [186], the digital-based ECG compression methods can be classified in: (i) direct methods, (ii) parameter extraction methods, and (iii) transform domain methods.

For example, Dallet *et al.* [226] proposed a specific ADC designed for performing ECG compression in hardware. In this case, no additional computational load is performed by the microcontroller embedded in the ECG data acquisition system in performing the compression.

The hardware-based solutions require the development of specific hardware that must be integrated into the acquisition device. For this reason, in several cases, a digital-based solution is preferred.

Direct methods perform compression by removing the redundancy of the ECG signal in the time domain [164]. Parameter extraction methods are based on the extraction of some features of the ECG signal (e.g. P wave, T wave, QRS complex) and on providing a compressed version of the signal according to these features [251]. In transform domain methods, the ECG signal is projected to a transform domain by means of a linear orthogonal transformation. In this case, the obtained coefficients are encoded to provide the compressed data [39].

The transform domain methods have gained significant attention due to their good capability of representing the ECG signal even at high compression ratios [186]. However, most of them require a high computational load to be implemented in real-time on data acquisition systems having low resources [186].

Alternatively, Compressed Sensing (CS) has been proposed in the literature for ECG data compression [186]. The advantage of CS — as compared to other methods — relies in its capability of achieving performance comparable with the transform-domain methods, while moving the computational load from the data

acquisition system to the node that receives the compressed samples. Thus, this solution has been widely used for implementing data compression on devices with constrained resources, such as wearable devices.

In some cases, the data rate reduction can be obtained by optimizing the resolution of the data, thus introducing a controlled quantization [111, 18].

As compression algorithm for the approaches further described in this thesis, we have chosen a simple 1-bit quantization, as proposed by Picariello *et al.* [186]. The algorithm is applied to heartbeat signals. During this phase, (i) the data is normalized, (ii) the dither is applied, and (iii) the 1-bit quantization is performed. With *normalization* we simply refer to the application of the formula

$$hbs_i = \frac{hbs_i - min(hbs)}{max(hbs) - min(hbs)}$$

for each sample i of the heartbeat signal hbs; in other words, we normalize the data between the minimum and the maximum value for every heartbeat signal. As a result, all the values will be in the interval $[0, 1]$.

The *application of dither* consists in applying a Gaussian dithering noise to the heartbeat signals with power σ. Thus, let be (i) hbs a heartbeat signal, (ii) ds the noising signal obtained by imposing $ds_i = \sigma \times random(0, 1)$ so that $|ds| = |hbs|$. The noised version of the original signal is given by $nhbs = hbs + ds$. Dither can be considered as a kind of noise, but it is typically and intentionally applied to randomize quantization error and thus to improve the next quantization step [188].

Finally, *1-bit quantization* step performs a comparison with a pre-defined threshold γ: if a given sample value $nhbs_i$ exceeds the threshold γ, 1 is assigned to the output vector, while 0 is assigned otherwise. Formally, for each $nhbs_i$:

$$qhbs_i = \begin{cases} 1, & \text{if } nhbs_i \geq \gamma \\ 0, & \text{otherwise} \end{cases}$$

Part II

Biomedical Monitoring of Drivers

CHAPTER 4

Biometry Definition

Contents

4.1 Introduction

When carrying out a task requiring high mental effort, the main difference between a professional driver and another person of the same age and physical constitution is the ability of the driver to be much *more economical* in terms of mental energy consumed [21, 22, 20]. Such a result led Formula Medicine[1] to the

[1] http://www.formulamedicine.com/en/

development of tools and methodologies for a highly sophisticated and specific mental training, namely *Mental Economy Training* (MET).

MET is an innovative training system aiming at optimizing an athlete's psychophysical resources, which also reflects an improvement of athlete's performances. MET finds its basis in the collection of biometric and physiological data, correlated to the execution of specific tests and exercises (*e.g.,* the classical strop game [220]). This process allows to deduce particular conditions and attitudes of a person, as well as her limits and the areas of intervention, in order to correct those behaviors which may cause a risky activity, *e.g.,* stress, excessive mental energy consumption, lack of concentration.

The pre-requisite of an effective MET session is the continuous monitoring of biometric and physiological data. Such a monitoring during a training session produces a huge amount of data to be analyzed, even for a short session. Indeed, many signals need to be acquired for an effective MET session, such as the Heart Rate (with sampling frequency of 1 Hz) and all the brain-related traces: the alpha, high-beta, low-beta, delta, theta, gamma, raw and attention signal, acquired with a sampling frequency of 100 Hz. This makes the identification of the above issues a time consuming and error-prone task. In addition, this inhibits the scaling-up of the MET, *i.e.,* the training of a large number of people without involving also a large number of trainers.

The contribution of this PhD work—during the period in Formula Medicine—was mainly dedicated to the support in the definition of novel metrics dedicated to the objective measurement of a performance while conducting a mental test.

4.2 MET: Mental Economy Training

This section describes the Mental Economy Training. Before presenting the process and tools behind such a training, we present the results of a study that motivates the need to have a mental training besides a physical training.

4.2.1 The Optimization of Brain Resources

Even the greatest champions of Formula 1 are unable to drive a whole race on the limit as they do when qualifying. Normally their race pace is about three tenths of a second per lap less than the maximum. Only when race strategy requires them to push hard they can reach the limit, but only for a limited number of laps. This means that every driver has within himself the potential to improve his performance over an entire race by about 20/25 seconds, a huge improvement in the world of motor sport where most of the time improvements are measured in milliseconds. Of course, this means that the driver should be able to keep her maximum for the whole duration of the race.

When pushing hard the heart rate of drivers increases by about 15-20 beats per minute compared to that measured during normal race pace. This means that to make up three tenths of a second per lap causes a significant increase in psycho-physical stress. The conjecture is that the increased effort is almost entirely attributable to a higher use of the nervous system and ultimately of the brain. And it is for this reason that a driver is unable to drive an entire race at maximum effort as in qualifying, as she would be liable to premature mental exhaustion.

To demonstrate such a conjecture, Bernardi *et al.* [21, 22, 20] conducted a study to investigate the characteristics that distinguish the brain of a professional driver. Specifically, twelve professional drivers, most of them coming from Formula 1, were subjected to intense mental tests (*e.g.,* reaction time tests) during which their brain performance and expenditure of mental energy was examined using Functional Magnetic Resonance Imaging (see Figure 4.1).

The performance achieved by professional drivers was then compared with a group of twelve non-athletes of the same age and (more important) same physical characteristics. The comparison debunked the anecdotal evidence that Formula 1 drivers are, in some way, particular men. In fact, judging only mental performance, there was no difference in the results between the drivers and the group of "normal" people. However, a deeper analysis of the results indicate that drivers' brains had a greater optimization of their resources and used fewer areas of the brain (see Figure 4.2). These areas were seen to be engaged at a much lower level while also having a faster and more efficient communication between the areas of

Figure 4.1: Mental energy examination

the brain used. In other words, for equal performance, the drivers' brains were much more economic in terms of expenditure of neurological energy. This suggested that better performance can be achieved by training people to optimize their brain resources when performing high demanding tasks.

4.2.2 The Training Process

Based on the results achieved by Bernardi *et al.* [21, 22, 20], Formula Medicine devised the Mental Economy Training (MET), a set of tools and techniques to evaluate objectively both the performance of the brain and expenditure of neurological energy. The goal is to train athletes to continually increase their mental performance while continually lowering their expenditure of neurological energy, a feature that distinguishes the champions of motorsport.

The MET has an *a-priori*-defined process, which is composed of three different phases. The first phase (**Analysis**) aims at objectively studying the individual's psychological profile and attitudinal traits. Tests are used to objectively analyze the individual's psycho-physiological features, such as *predisposition factor* (the ability to keep the mind free and muscles relaxed), *performance factors* (mental performance measured with concentration, attention focus, multitasking, strategy ability tests), and *personality factors* (individual traits: self-confidence, anxiety management ability, rationality, hyper-control). This is a fundamental phase to understand the areas of intervention to improve her performance.

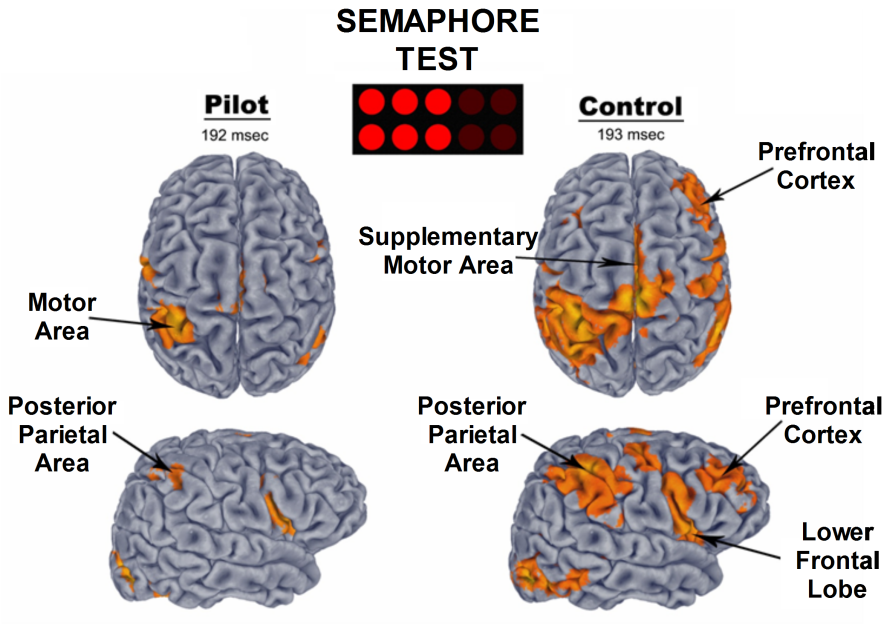

Figure 4.2: FMRI results. The picture shows the areas of brain activated by drivers and "normal" people during the semaphore test, a test aiming at measuring the reaction time of the trainee.

In the second phase (**Self-Awareness**), the trainee acquires awareness of her limits. Tests are used to increase the sensitivity of the information coming from the body and the mind, even during a task. Tools for improvement are the result of the interaction between trainer and the individual, who plays an active role. It is only by acquiring the ability to analyze herself that the trainee can improve her performance.

Finally, in the last phase (**Optimization**) the trainee can improve her performance with simultaneous reduction in energy consumption, even under stress. Tests are used to reach maximum performance with minimum psychophysical consumption (e.g., mind kept free from distracting/useless thoughts, extreme control of muscle tension) even in conditions of stress. Some effective advantages are the ability to manage psychological pressure and stress, better decision-making ability, reduced risk of burn-out.

Figure 4.3: Coordination & Body Tension Control Test

The MET process is conducted through several gamified tests. One of the most famous is represented by the classical stroop game or color test [220]. During this test, subjects are required to answer to several question concerning the colour of the words appearing on the screen. For example, if the word "BLUE" is printed in a red color, the user should opt for "FALSE", otherwise if the word is coloured bu the same colour that it represents, the user should opt for "TRUE". Thus, the color test is a concentration test where ability to clear the mind, limit the frontal cortex activity, reduce thoughtful activity, isolate from any mistakes are needed requirements.

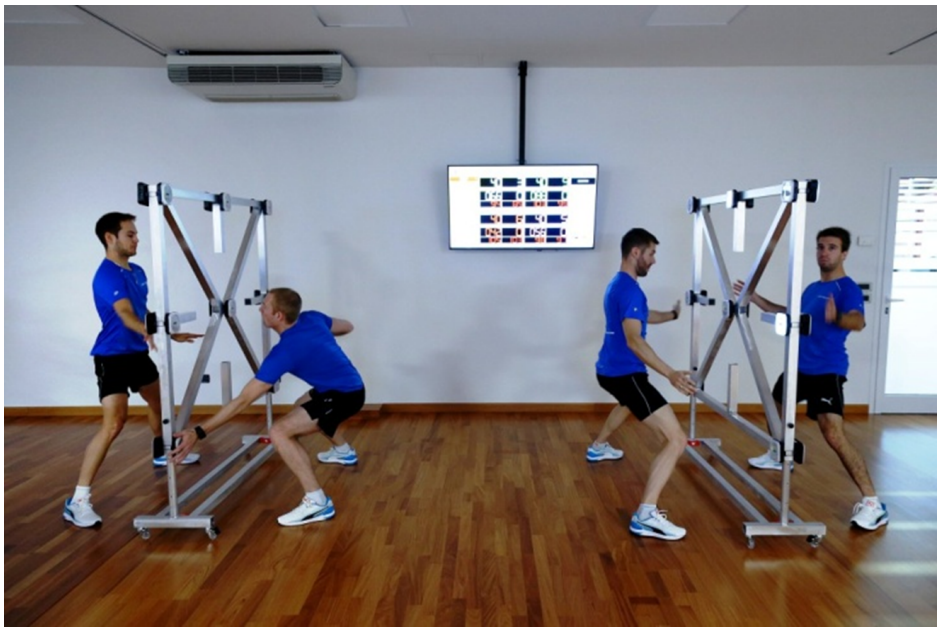

Figure 4.4: Neuromuscular Agility Test

A second example of training test is represented by the "Coordination & Body Tension Control" test (see Figure 4.3). It requires to maintain balance on an unstable board for 60 seconds while watching the result on the monitor (visual feedback). The result is related to the capacity to control muscle tension.

The "Neuromuscular Agility Test" is another test of a MET session (see Figure 4.4). Physically it is composed of 8 lights mounted on a square frame. Each light has a sensor in its center which works as an off switch. The training consists in turning lights off as fast as possible by covering the sensor with the palm of the hand. It is necessary to stand at the correct distance from the support frame. The user's goal is to switch off as many lights as possible in 60 seconds.

The MET process is a complex process that requires a continuous monitoring of biometric and physiological data during each test. From the analysis of this data, it is possible to (i) evaluate the effectiveness of the training and (ii) change the training protocol if needed. The huge amount of data to analyze requires

very often a one-to-one relation between trainer and trainee. This represents a threat to the scalability of the project.

4.3 Objective Measurement of a Performance

Through the evaluation of metrics, indices have been defined with the aim at objectively measuring the performance of a subject involved in a gamified version of the *stroop test* [221]. This test involves colored words appearing on the monitor in rapid succession. Below the colored word are two boxes, containing the words *true* and *false*. The subject has a gamepad in which only the top two buttons are active: the left one corresponds to the left pane, the right one to the right pane. If the color and word match, the subject must press the button that matches the true box. If not, the user must press the button that corresponds to the false box. True and false can alternate randomly between the left and right boxes. The *timeout* is a parameter that handles the timing within which the subject must provide the answer.

The indices—defined and experimented within the present PhD project—have been specifically designed to provide an objective measure of the performance of an athlete engaged in this gamified variant of the stroop test. The metrics that return the indices are summarized as follows:

- *Mental Economy Index*: is a global index, i.e., a number that summarizes a performance in terms of test results, brain data, and cardiological data. In particular, the metric that allows the calculation of this index, incorporates the information on the average response time, the result in terms of percentage of correct answers, the trend of the brain wave related to the attention level and the trend of the heart rate.

- *Integrated Performance*: is a local index, that is, it summarizes the performance only based on the test results. In detail, the calculation involves weighting the average response time with the result obtained.

- *Psychophysical Economy*: is a full-body index that describes a performance in terms of test results, brain data, and cardiology data. The metric con-

siders characteristics assessed on the trend of the attention curve and heart rate over time.

- *Pure Performance*:in this case, the performance related to the test result is integrated with the difficulty of the timeout. The two variables are weighted averaged, giving more weight to the result obtained in terms of correct answers.

- *Reactivity*: a percentage value describing the subject's responsiveness in providing answers. The scale of percentage values has been defined ad-hoc. For example, the latter takes into account that, from the value of 5 seconds up to the threshold value of 1.6 seconds, the percentage assigned to performance can have 50 % as the maximum value. Below 1.6 seconds and up to the lower limit of 0.5 seconds there are all the percentages that describe good responsiveness.

- *Tension Handling*: the calculation is based on the weighting of specific *features* derived from the heart rate waveform. These are the mean, trend and oscillations weighted by 40 %, 30 % and 30 % respectively. This indicator is obtained from a metric that incorporates information such as mean, trend and oscillations obtained from the attention curve. These magnitudes are weighted by 70 %, 20 % and 10 %, respectively.

With reference to these indicators, it is considered necessary to emphasize that a performance is considered optimal if the results —trictly related to the test—show no error and if the cardiac and cerebral data agree in recording a constant trend (therefore obtained with minimal psychophysical effort).

In Figure 4.5 is shown the industrial software with a preliminary version of the Mental Economy Index.

Figure 4.5: A preliminary version of the Mental Economy Index integrated in the software of Formula Medicine. The index is represent by the speedometer at the bottom of the image.

CHAPTER 5

Integration of Biometry and Telemetry

Contents

5.1 Introduction

The two partners of this PhD project, the Italian company Formula Medicine and the French company AOTech, together with the University of Molise have undertaken a work of research and experimentation with the aim at studying

potential correlation between the body control skills and the body parameters of drivers during a car race.

Together with a dedicated team of AOTech engineers, we initially defined the following main work activities:

- study a way to integrate the body sensors with the AOTech simulator without encountering any inconvenience on the data;
- collect several sample of telemetry data.

After these preliminary studies, we worked at the definition of a tool which allowed the offline data integration between the AOTech simulator (telemetry) and the Formula Medicine body (biometry) sensor systems. Thus, it is a tool that allows the analysis of data regarding the race and the body of drivers, by offering a contextualization of how a driver is capable of control his body during a race. The tool had to offer also a graphic section where the experts can potentially view the details for each of the involved data sources of information. Finally, our tool had to integrate a Decision Support System with the aim at supporting the trainer during a car race, by offering automatic analysis of potentially dangerous episodes thanks to the involvement of statistical learning techniques.

5.2 Research Motivations

Without a proper data integration system, an assistant or a trainer of professional drivers can analyze and judge the performances of his trainee based only on the data obtained by the telemetry of the car or on the data obtained by the network of body sensors. With our system, we aimed at offering the chance to assist professional drivers in terms of vehicle and body parameters. Our aim, thus, was to create a software tool capable of:

- reading the data from the two involved systems;
- performing the data integration and synchronization;
- offering a visualization section;
- providing typical Decision Support System functionalities.

In this way, a trainer of a professional driver, may have a complete and detailed view on how the driver is facing every moment of a race, both in terms of vehicle than body parameters.

5.3 Instrumentation & Acquisition Setup

The AOTech simulator system is shown in Figure 5.1[1]. The AOTech simulator offers the possibility to acquire and export data related to vehicle parameters.

Figure 5.1: The AOTech simulator system.

The body sensor network is basically composed of two independent instruments:

- the Brainco single electrode EEG-band[2];
- the Polar OH1[3].

[1] Courtesy of http://www.aotech.fr/simulator/
[2] https://www.brainco.tech/
[3] https://bit.ly/3h1UEBG

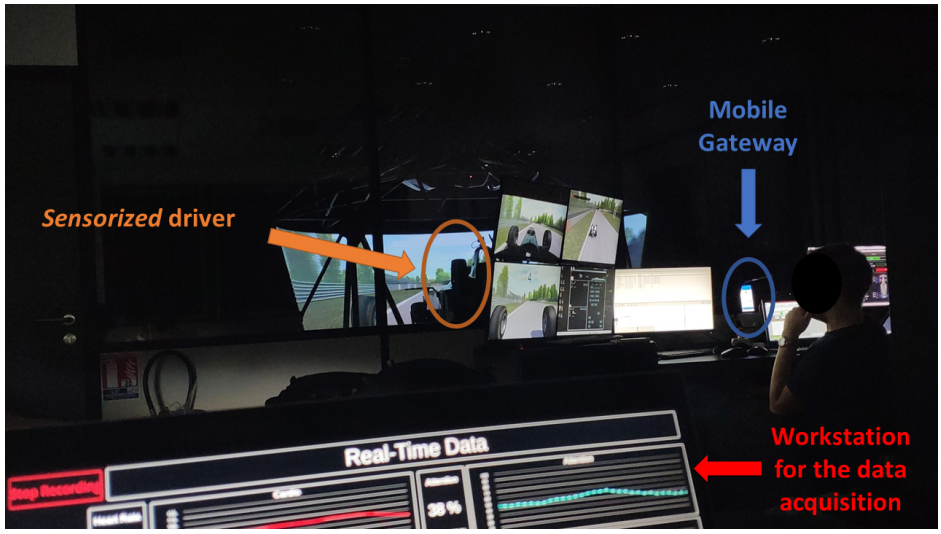

Figure 5.2: The instrumentation setup used for the acquisition of telemetry and biometry data.

Both these two sensors are connected thanks to a mobile gateway, developed on purpose. The acquisition setup can be depicted in Figure 5.2.

As it is possible to see from such a figure, a laptop was used as workstation for the receiving and storage of body parameters. These have been acquired through the mobile gateway that communicates — via Bluetooth low energy— with the sensors and — via Wi-fi — with the workstation.

5.4 The Data

The data was acquired through a typical car race protocol. Each driver had to conduct three training sessions: the first one—labeled as Free Practice—and the remaining two considered as Qualification. This difference was intended to provide an initial warm-up session for the driver before getting involved in the real training sessions. The difference only relies on the lasting of each record-

ing sessions. In details, a qualification recording session might cover around 20 minutes of simulated race.

Each of these files contains the following information:

- Time: it is an elapsed measure of time — expressed in seconds — which indexes each acquisition sampled at 50 Hz;
- CarSpeed: the speed of the vehicle in km/h;
- ACC_X, ACC_Y: acceleration samples acquired along the X and Y axis through an accelerometer installed onto the vehicle;
- pBrakeF: brake pressure applied on the front, expressed in bar;
- pBrakeR: brake pressure applied on the rear, measured at the rear master cylinder;
- SteerTorque: steering measurement, expressed as a steering angle.

On the other hand, each of these files—obtained from the body sensors—contains the following information:

- Number: an integer value which indexes each acquired row;
- Timestamp: acquisition timestamp expressed in the format YYYMMDDhhmmss[ms];
- playerId: an integer value that indicates to which player belongs the data;
- dataType: a string that specifies the type of the data. The admissible data types are:
  - *mindraw*, the raw data acquired by the frontal EEG electrode
  - *mindwaves*, the EEG data expressed in terms of the following frequency bands: α, δ, γ, high β, low β and θ;
  - *mindattention*, the EEG data expressed in terms of an integer value that indicates the attention value — on a scale from 0 to 100 — of the subject being monitored;

  - *heartrate*, an integer value indicating the heart rate of the monitored subject.

- dataValue: the field which explicits the value of the acquired samples.

No multimedia data, such as images or videos, was available for the elaboration.

5.5 The Proposed Trainer Support Tool

To explain the working principle of each of the functionalities offered by the tool, we are going to use real data from a sample acquisition obtained during the first period in AOTech.

First of all, the tool offers Data Parsing, Integration and Synchronization. The integration has consisted in an alignment in time of the several signals acquired from the two biometric sensors and the telemetry. This has involved a phase of resampling in order to integrate all the data. The synchronization has been automatically performed by agreeing a mutual instant for the beginning of the recording.

The visualization of biometry and telemetry data is offered, through the involvement of several graphs. Each lap has a dedicated graph, and every graph is composed of all the acquired waveforms, resampled to generate uniformly spaced data. An example of such a result is depicted in Figure 5.3.

Through the adoption of a differential analysis technique, our tool is capable of automatically select a potential error of the driver during the race. Such an error may regard—for example—the appearance of too many sudden and opposite steering movements. Indeed, by monitoring the steering torch data it is possible to evaluate when close episodes of an unexpected and fast variation of the acquired samples are detected. This can be better seen, circled in red, in Figure 5.4. This error is also accompanied by two contextualized local phenomena: an increasing of the heart rate and a decreasing of the attention level.

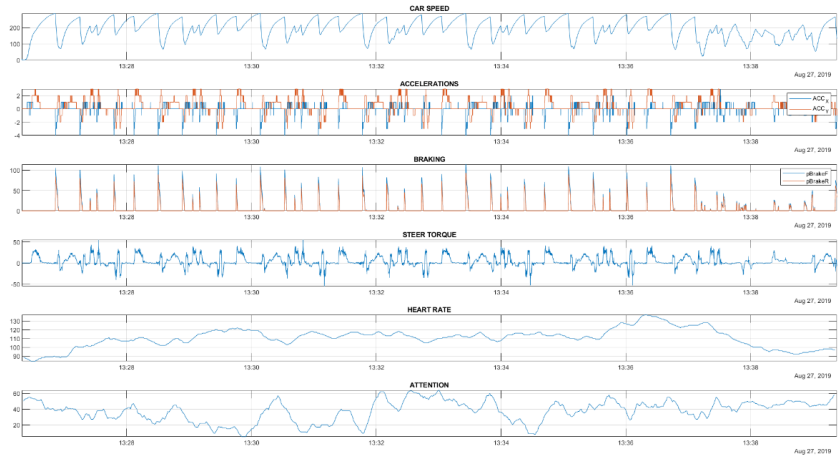

Figure 5.3: An example of biometry and telemetry data visualization offered by the trainer support tool.

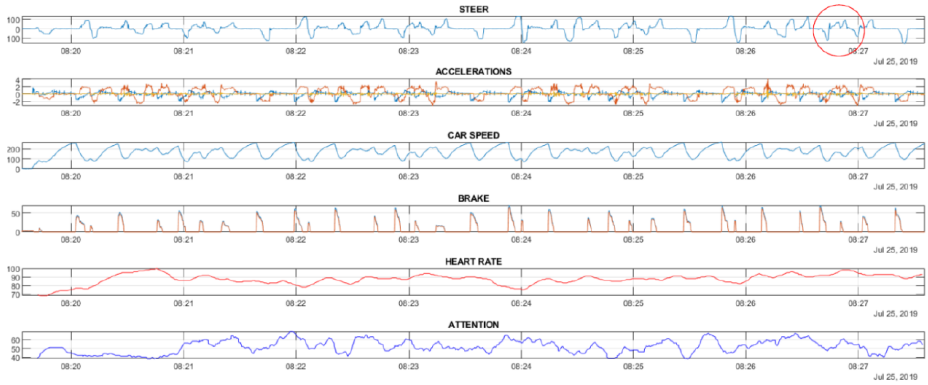

Figure 5.4: An example of an error performed by a driver highlighted on the biometry and telemetry data.

5.6 Final Remarks

At the moment of this thesis, the tool works as an off-line analyzer. It is composed of few Matlab files. There is no GUI or dashboard of any type, at the moment of the writing of this thesis. As future work, we intend to adapt it by offering the possibility to use it in real-time during a simulated race. However, many considerations can be done by having available such qualitative data. The contextualization between vehicle's parameters and body signals may lead to a deeper knowledge of how a driver is capable of managing his or her own body with respect to specific moments of a racing. Observing a simulated race with *an individual lap* level of granularity, a trainer can analyze the parameters acquired from both the systems even only in relation to the fastest and the slowest lap of the race that the driver had to face. For example, in Figure 5.5 is shown the fastest lap of a simulated race.

During this lap, the driver seems to conduct properly the control of the vehicle, according to the telemetry offered by the simulator. Moreover, during this lap, the driver's heart rate increases within an average of more than 10 beats per minute. Finally, the attention level decreases in average.

Instead, if the trainer would look with attention at the slowest lap (Figure 5.6) of the same recording session and for the same driver, a particular situation could be found.

In this case, especially in the second part of the lap, the car speed significantly decreases. And this occurs when the heart rate reduces of around 25 beats per minute. The last word about the chances that these phenomena may be a clue to a good or a bad behavior is let to the trainer. Our tool is intended as a support to the trainer for the decision-making process during the training of professional drivers by using the typical parameters available from the vehicle and two basic body signals, such as the attention level and the heart rate.

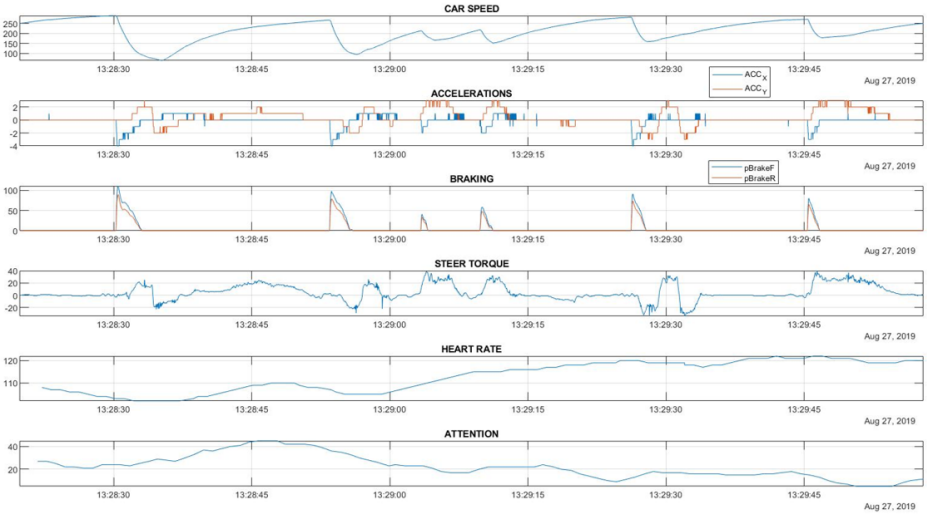

Figure 5.5: An example of fastest lap performed by a driver.

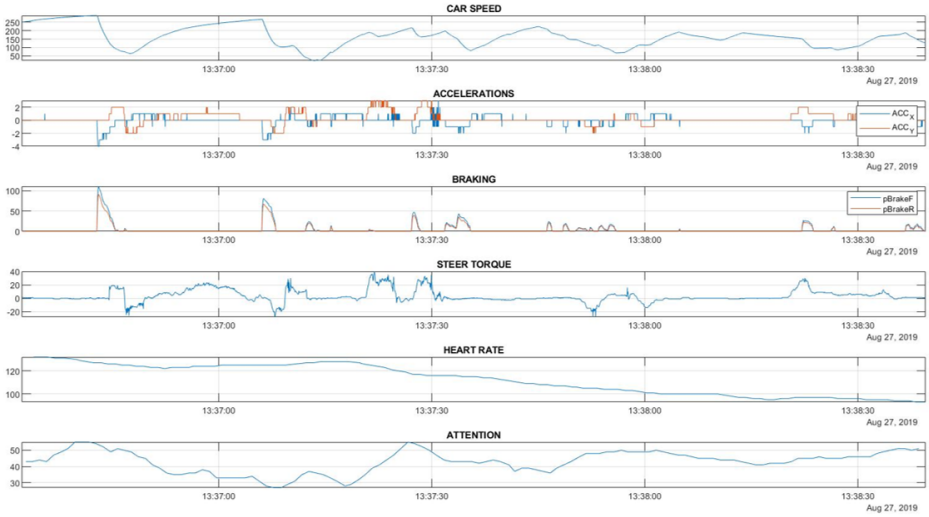

Figure 5.6: An example of slowest lap performed by a driver.

5.7 Wrapping-up

This is the last chapter dedicated the biometric monitoring of athletes during high demanding tasks. The further content of the thesis will extend the analysis of biometric data of athletes in the context of the continuous monitoring of an ECG trace. The continuous monitoring could be a very important aspect for athletes because—thanks to the automatic detection of potential anomalies—it can allow to provide support to medical stuff during the process of early diagnosis of pathological conditions.

CHAPTER 6

Monitoring of Health Status via ECG Analysis

Contents

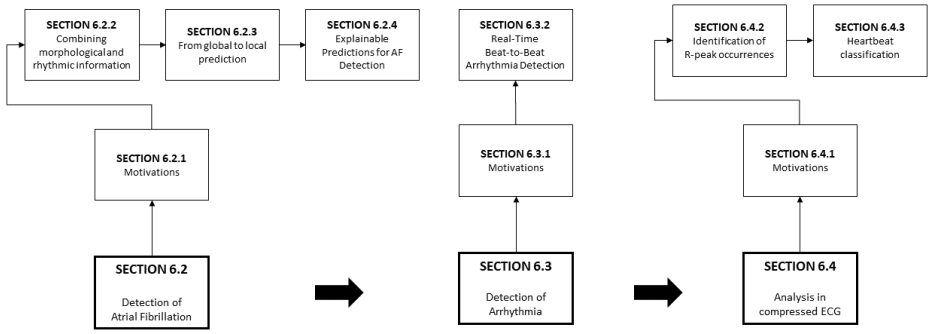

Figure 6.1: A road-map of the chapter.

6.1 Introduction

The continuous monitoring of—at least— a single digital ECG signal allows to design a system for the medical support. Indeed, the real-time disposal of such data becomes crucial in the contexts of early diagnosis of pathological conditions. In this chapter, the approaches for the detection of several pathologies are presented. To support the reading of the present chapter, a specific road-map is offered in Figure 6.1.

In particular, this section concerns with three approaches for the automatic detection of Atrial Fibrillation episodes, one approach for the automatic identification of different arrhythmia conditions and two approaches of information retrieval in compressed ECG.

6.2 Detection of Atrial Fibrillation

In this section, three approaches for the automatic detection of Atrial Fibrillation events are presented. The first one, namely *MORPHYTHM*, combines two sources of information from a digital ECG signal: the rhythmic and morphological one. The combination of such features together with novel metrics defined in this context has allowed to define an approach that has shown imprevements in two crucial aspects of the medical binary classification: the reduction of False Negatives and the increase of True Positive. Then, we ideated a new approach,

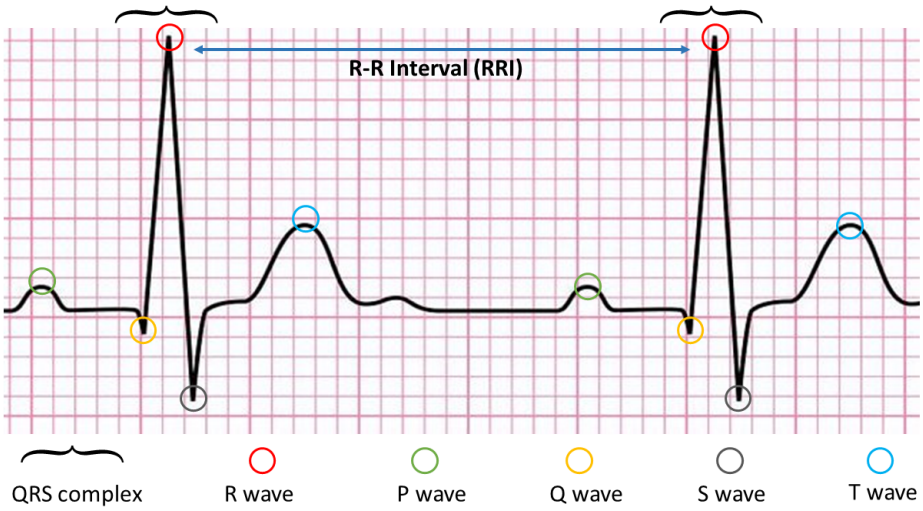

Figure 6.2: ECG theoretical waveform.

namely *LOCAL MORPHYTHM*, where we use some of the results obtained from *MORPHYTHM* to build an automatic local approach, e.g. with a refined training of the Machine Learning model. Finally, we have observed that the combination of morphological and rhythmic information does not allow to produce an explainable approach. For this reason, we ideated *AMELIA*, a method that is inspired by the manual observation of cardiologists. With this approach, we aimed at providing—despite the detection—explainable information on the causes of the anomaly.

6.2.1 Motivations

According to the official international guidelines [124], AF can be detected by observing three main features in the ECG, as shown in Figure 6.2, *i.e.,* (i) absence of the P-wave, (ii) presence of fluctuating waveforms (f-waves) instead of the P-wave, and (iii) heart rate irregularity. The first two features can be defined as *morphological*, while the third one is *rhythmic*.

Since AF is often asymptomatic [120, 30], a reliable device combined with an accurate, real-time, and automatic AF detection algorithm is desirable for improving detection of AF [120, 30, 32, 33].

Normally, the heart contracts and relaxes to a regular beat. In atrial fibrillation, the upper chambers of the heart (the atria) beat irregularly (quiver) instead of beating effectively to move blood into the ventricles[1]. If the ECG recording is available, AF is diagnosed by whenever an irregular heartbeat presents the following characteristics: the absence of P waves (with disorganized electrical activity in their place) and irregular R–R intervals due to irregular conduction of impulses to the ventricles [75].

The prevalence of atrial fibrillation is increasing all over the world and it is becoming one of the most important clinical issues for industrialized countries [252, 75]. AF is a crucial risk factor for the occurrence of stroke. Beyond stroke, AF can lead also to congestive heart failure. Furthermore, hypertension, diabetes and heart failure are some of the most common comorbidities [142, 219]. In addition, AF presents a sever influence on the global health conditions of individuals who contract it [125].

To produce a diagnosis of AF, a cardiologist checks the clinical history of the patient and the ECG signal, by at least observing a single lead during the revealing of the episode [75]. Unfortunately, AF is often paroxysmal, *i.e.,* there are recurrent episodes that stop on their own in less than seven days[75], and asymptomatic. For these reasons, the screening of such a pathology needs to become a priority.

Usage Scenarios of AF Detectors

AF detection methods might be useful in two different scenarios: *offline* and *online*. In an *offline* scenario, the ECG of a patient is recorded and, later, an AF detection method is used to find possible AF events occurred in the recorded period. This can help physicians discovering AF events in possibly long EGCs.

AF detection methods could be also particularly valuable in an *online* scenario: while the ECG is acquired, it is immediately passed to the AF detector,

[1] https://bit.ly/3dvrXJX

which promptly detects AF events. Online AF detection can be useful in tele-medical applications, where patients are constantly monitored.

The application context we consider is the *online* monitoring. In other words, we assume that we have chunks of ECG incrementally available. Therefore, we specifically focus on real-time (or near real-time) approaches. Online monitoring is useful for tele-medical applications.

Several tele-medical projects were proposed in the literature. Zhu *et al.* [250] introduced the SPHERE system, which combines several sensors which acquire data through wearable, environment, and video devices. Villar *et al.* [231] introduced Hexoskin, a line of cutting-edge smart clothing that include body sensors into garments for health monitoring. Balestrieri *et al.* [11] recently introduced ATTICUS, an innovative Internet of Medical Things (IoMT) system for implementing personalized health services.

6.2.2 Combining Morphological and Rhythmic Information

In the last decade, several methods have been pro-posed for the automatic detection of AF. Most of them have shown good results by exploiting only the analysis of heartbeat rhythm. Therefore, in this context, room for improvement could be filled with an automatic approach having a combined knowledge. Thus, in this section, we present *MORPHYTHM*, an approach based on machine learning techniques that combines rhythmic and morphological features to detect AF events. *MORPHYTHM* uses the most promising state of the art rhythmic and morphological features and some novel features.

Preprocessing of ECG Data Before extracting features, the ECG data (from the AFDB) have to be pre-processed according to works in [178] and [42]. The main steps involved in this phase are the followings:

- **Detrend of ECG signal**: the offset has been removed from the raw ECG signal by removing the mean from the signal.

- **Filtering Stage**: first, a low and high pass filters have been applied to get rid of baseline wander and discard high frequency noise, respectively.

Subsequently, a derivative filtering has been operated on the signal aiming at emphasizing the high frequency components of the ECG.

- **Sample Amplitude Normalization**: each recording has been normalized in terms of sample amplitude around the maximum.

Rhythmic Features Rhythmic features are based on one or more heartbeats and they aim at capturing aspects that mostly regard the regularity of the heartbeat signal. Zhou (2015) *et al.* [249] state that the detection methods based on RRI are more useful to produce a precise and accurate identification of AF because the R-wave peak of the QRS complex is the most prominent characteristic feature of an ECG recording. Such a characteristic is less subject to noise [149, 134, 145, 103].

In *MORPHYTHM* we use two features based on the observation of a single heartbeat signal, *i.e.,* HBL and HBDL, and two additional rhythmic features that consider the information of a sequence of consecutive heartbeats, *i.e.,* HBR and Entropy:

- **heartbeat Length (HBL).** This feature represents how long a single heartbeat signal lasts. We measure *HBL* as the number of samples from a peak R to the next peak R;

- **heartbeat Discrete Length (HBDL)**. Such a feature is a classification of the heartbeat signal in three classes, based on its length. A beat is (i) *short* if it takes less than 0.5 seconds, (ii) *long* if it takes more than 1.2 seconds, and (iii) *regular* otherwise;

- **heartbeat Regularity (HBR)**. This feature is based on HBDL. It considers a rhythmic pattern of 10 consecutive discrete heartbeats lengths. Once obtained the pattern, we compute HBR simply counting the number of regular heartbeats. It is worth noting that there are approaches in the literature which consider a very short windowed sequence of heartbeats [184];

- **Entropy**, as defined by Zhou (2015) *et al.* [249] and described in the background section dedicated to the baseline work chosen for Atrial Fibrillation detection.

While HBL and Entropy have been previously used in the literature [249], HBDL and HBR are two new rhythmic features defined with this method.

Morphological Features Even if the acquisition of rhythmic features can be very reliable, such features can only help detecting arrhythmia, which is just one of the possible signs of AF. On the other hand, morphological features are necessary to detect anomalies in the shape of a single heartbeat signal.

In *MORPHYTHM* we propose three different measures that—given a sequence of samples provided for a heartbeat signal[2]—return a single numeric value:

- **Mean Signal Intensity (MSI)**. Such a feature is measured as the mean of all the samples acquired in a heartbeat signal. The mean signal intensity, alone, provides a very rough indication of regularity of the heartbeat signal. If there is any anomaly in any part of the heartbeat signal, such a feature may help identifying it. For example, if the P-wave is missing, the MSI may be slightly affected;

- **Signal Intensity Variance (SIV)**. This feature is measured as the variance of all the samples acquired in a heartbeat signal. The SIV helps characterizing the heartbeat signal: again, a low SIV might indicate the absence of the P-wave.

- **Signal Intensity Entropy (SIE)**. This feature is computed as the entropy [157] of the distribution of the sample values in a heartbeat signal. This feature is similar to SIV, *i.e.,* it is aimed at representing the variations in the signal of a heartbeat.

It is worth noting that extracting features by considering a whole heartbeat might compress too much the information in the ECG data. To extract richer

[2]We consider as a *heartbeat* a digital signal which goes from a R-peak to the successive. Such an interpretation is very suitable for AF detection, because it highlights the atrial activity.

information, we also propose a novel descriptor of a heartbeat signal by (i) dividing the whole heartbeat in n segments; and (ii) computing the above defined features on each segment:

- **Segmented Mean Signal Intensity (S-MSI$_i$)**: given the i-th segment of the heartbeat signal, S-MSI$_i$ is computed as the mean of the sample values of such a segment.

- **Segmented Signal Intensity Variance (S-SIV$_i$)**: given the i-th segment of the heartbeat signal, S-SIV$_i$ is computed as the variance of the sample values of such a segment.

- **Segmented Signal Intensity Variance (S-SIE$_i$)**: given the i-th segment of the heartbeat signal, S-SIE$_i$ is computed as the entropy [157] of the sample values of such a segment.

All such features allow to roughly represent the shape of the signal of the heartbeat. We reduce the resolution of the heartbeat signal to just 30 values (n=10 for each feature) to reduce the noise of the samples.

Besides the aforementioned features, we also integrate in *MORPHYTHM* other state of the art morphological features:

- **Fast Fourier Transform (FFT$_i$)**: we include the features introduced by Haque *et al.* [96] by calculating the Fast Fourier Transform of the heartbeat signal on 32 points.

- **Auto-Regressive Model (ARM$_i$)**: we include the features introduced by Zhao *et al.* [247] by estimating the coefficients of the Auto-Regressive model of order 16.

Putting all Together *MORPHYTHM* combines all the features we previously described using supervised machine learning techniques. After the training phase, *MORPHYTHM* is able—given a heartbeat signal—to classify it as *fibrillating* or *not fibrillating*. In the *MORPHYTHM* evaluation, we experimented several classifiers.

Empirical Evaluation

The *goal* of this study is to evaluate the accuracy of *MORPHYTHM* is classifying AF events in a patient. The *perspective* is both (i) of a researcher who wants to understand if combining rhythmic and morphological features is useful for detecting AF events, and (ii) of a practitioner who wants to use the most accurate and precise approach in a telemedicine application. Thus, the study is steered by the following research question:

Can the combination of rhythmic and morphological features improve the classification accuracy of Atrial Fibrillation events?

Context Selection The context of this study is represented by MIT-BIH AF Database [82], a commonly used benchmark which contains recordings of 25 patients. Due to the embedding of morphology descriptors, our overall study has been performed on the $AFDB_1$, *i.e.,* the AFDB without records 00735 and 03665 because, for such records, only information on the rhythm is available [82]. Each recording in the dataset lasts 10 hours and contains two ECG signals sampled at 250 samples per second (12-bit resolution).

In the context of our study, we experimented a large set of machine learning technique to train *MORPHYTHM*. Especially, we experimented tree-based classifiers, *i.e.,* J48 [190], Replication Tree [60], and Random Forest [16]. Such approaches, indeed, can build models that are also easy to understand by a human. We also experimented Logistic regression [50] and AdaBoost M1 [69].

Experimental Procedure To evaluate the accuracy of *MORPHYTHM*, we used a classical Leave-1-Person Out (L1PO) cross-validation: we divided all the data in n folds, one for each patient, and we use one at a time each of such folds as test set and the union of the remaining folds as training set. This means that the data related to a single patient were embedded once in the test dataset and *n-1* times in the training dataset. This technique allows to build a classifier which is not trained and tested on the data belonging to the same patient. We did this to evaluate the technique in the most challenging scenario: the ECG of different patients can be very different.

We compared *MORPHYTHM* to the approach proposed by Zhou (2015) *et al.* [249], previously presented in the background section of this thesis dedicated to the AF detection. The instances to be classified were all the single heartbeat signals provided in the dataset, labeled as *fibrillating* or *non-fibrillating*. The work by Zhou (2015) *et al.* [249] just reported the performance of the approach globally, *i.e.,* for all the patients. Instead, we provide the performance of the approaches with a finer grain, *i.e.,* on patient-by-patient base. Since we do not have the patient-by-patient results for the baseline, it was necessary to re-implement the approach and to re-compute the results.

To answer our research question, we compared two critical aspects: True Positives (TP), *i.e.,* the number of instances classified as *fibrillating* by the approach and that were actually *fibrillating*, and the False Negatives (FN), *i.e.,* the number of instances classified as *non-fibrillating* which were, actually, *fibrillating*. A high number of TP is desirable, because it indicates the number of AF episodes correctly detected. Also, ideally, a perfect approach does not lose any AF episode: thus, keeping the number of FN low is very important.

We use a Wilcoxon signed-rank test to verify if *MORPHYTHM* achieves statistically significant better results than the approach proposed by Zhou (2015) *et al.* [249]. To do this, we use the results achieved patient by patient in terms of TP and FN. Formally, our null hypotheses are:

- $H0_1$: *MORPHYTHM does not identify a higher number of TP as compared to the approach proposed by Zhou (2015) et al. [249]*;

- $H0_2$: *MORPHYTHM does not identify a lower number of FN as compared to the approach proposed by Zhou (2015) et al. [249]*;

We reject a null hypothesis if the p-value is lower than $\alpha = 0.05$.

Even if we evaluate the possible improvement only on TP and FN, we also report the global results in terms of True Negatives (TN — *i.e.,* instances correctly classified as *non-fibrillating*) and False Positives (FP — *i.e.,* instances classified as *fibrillating* that are, actually, *non-fibrillating*).

Table 6.1: Comparison of *MORPHYTHM* and the approach proposed by Zhou (2015) *et al.* [249]. In boldface the results achieved by *MORPHYTHM* that are better than the baseline.

Approach	**TP**	**TN**	**FP**	**FN**	**ΔTP**	**ΔFN**
Zhou (2015) *et al.* [249]	489,834	603,216	17,188	19,911		
MORPHYTHM — Random Forest	490,810	584,692	35,612	18,935	**+976**	**-976**
MORPHYTHM — J48	479,411	560,567	57,049	33,122	-10,423	+13,211
MORPHYTHM — Logistic	494,255	595,664	24,789	15,445	**+4,421**	**-4,466**
MORPHYTHM — AdaBoost M1	494,384	601,974	18,430	22,362	**+4,550**	+2,451
MORPHYTHM — RepTree	481,397	571,262	49,142	32,348	-8,437	+12,437

Analysis of the Results

We show the global performance of the compared approaches in Table 6.1. For *MORPHYTHM*, we also specifically report the difference in terms of TP and FN with the baseline, *i.e.,* ΔTP (the higher, the better) and ΔFN (the lower, the better), and we put in boldface the cases in which *MORPHYTHM* achieves better results.

The first consideration that can be derived from the analysis of Table 6.1 is that three (*Random Forest, Logistic and AdaBoost M1*) of the five chosen machine learning are able to achieve better results than the baseline. Furthermore, the Logistic method performs definitely better than its competitors by showing an improvement of around 4,400 heartbeats compared to the method by Zhou (2015) *et al.* [249]. If we would assign an inter-beat interval of 0.5 seconds, an improvement of 4,400 indicates more than *35 minutes* of AF rhythm improved in the classification with respect to the baseline. Even if this could appear as a negligible result, it should be noticed that the accuracy level achieved by such approaches is very high and, therefore, even achieving a small improvement is very difficult.

It can be noticed that the global accuracy of the approach by Zhou (2015) *et al.* [249] slightly differs from the global accuracy reported in the original paper. Especially, in the original paper the authors reported the following values for *sensitivity*, *specificity*, and *accuracy*—they just report aggregated measures: 97.31%, 98.28%, and 97.89%. With our replication of the approach by Zhou

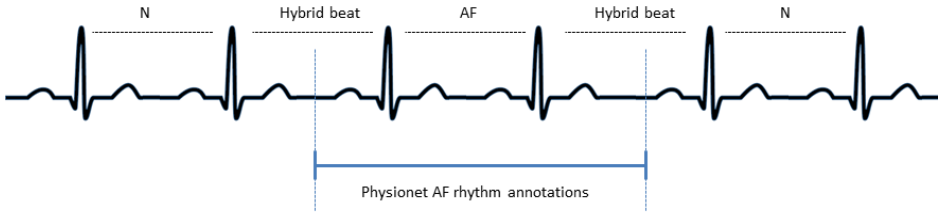

Figure 6.3: A graphical example of hybrid heartbeats.

(2015) *et al.* [249] we achieve the following results: 96.09% of sensitivity, 97.22% of specificity, and 96.71% of accuracy. We are confident that the different results are not due to implementation errors, but to different choices in the evaluation design. The different results could be due to the following reasons:

- the transient: to avoid any error due to interpretation, the first 128 (126 coming from the entropy compression + 2 from the word sequence evaluation) beats have not been considered in the replication of the work by Zhou (2015) *et al.* [249]. Unfortunately, in the paper by Zhou (2015) *et al.* [249] there is no clear indication on how the authors deal with the initial 128 beats;

- the timestamps: Physionet offers two different timestamps: one for each beat classification another one for each AF events (rhythm annotation). There are cases where there is a mismatch between the two timestamps, *i.e.,*, the AF event does not start (or does not end) with the beginning of a (or the end) of beat. In other words, beats and AF events are not always synchronized. This causes an ambiguity regarding the interpretation of "hybrid beats", *i.e.,* beats that are not aligned with an AF event (see Figure 6.3).

It is worth noting that the different results achieved does not represent a threat for the final message of this method. Indeed, improving the accuracy of the approach by Zhou (2015) *et al.* [249] by performing a different evaluation design likely results in an improvement of *MORPHYTHM* as well, since the approach by Zhou (2015) *et al.* [249] is one of the features exploited by *MORPHYTHM*.

Table 6.2 shows a patient-by-patient comparison between the approach by Zhou (2015) *et al.* [249] and *MORPHYTHM* (with the classifier that achieves the best results globally, *i.e.,* Logistic Regression). Even if the method proposed by Zhou (2015) *et al.* [249] is incredibly accurate, as it can be observed from Table 6.2, *MORPHYTHM* achieves much better results for some patients and comparable results on some other patients.

Specifically, *MORPHYTHM* identifies a higher number of TPs for 15 out of 23 patients and it identifies a lower number of FNs for 15 out of 23 patients. The results of the Wilcoxon signed-rank test show that we can reject both our null hypotheses: *MORPHYTHM* identifies a *significantly* higher number of TPs ($p = 0.021$) and a *significantly* lower number of FNs ($p = 0.014$).

Table 6.3 shows the comparison between *MORPHYTHM* and the approach by Zhou (2015) *et al.* [249] for one of the patients, *i.e.,* 05091. For such a patient, the baseline has never detected any *fibrillating* event. This means that this record has been classified as not affected by AF, overall, even if it was. On the other hand, most of the classifiers we consider are able to detect some *fibrillating* events. However, unfortunately, this is not true for the best classifier (*i.e.,* Logistic regression), which, similarly to the baseline, does not identify any heartbeat of the specific patient as *fibrillating*.

Final Remarks

The improvement achieved is promising; however, there is still much room for improving the accuracy. In order to increase the generalizability of our results, we aim at applying in the future at least one classifier for each family. In this work, for example, we did not consider Bayesian networks, Rules-based classifiers, and Neural Networks. Also, to maximize the accuracy of *MORPHYTHM*, it would be desirable to use feature selection, to remove useless features that could decrease the classification accuracy. Specifically, since morphological features can be very patient-dependent, it could be useful performing feature selection for each single patient rather than globally. Finally, we plan to perform a cost-benefit analysis. Indeed, in some online applications, it could be necessary to have some constraints, such as the total reduction of FN, even if the FP rate

Table 6.2: Patient-level comparison between *MORPHYTHM* and [249]. In boldface the best results for each patient.

Record	Zhou (2015) *et al.* [249]		*MORPHYTHM*	
	TP	**FN**	**TP**	**FN**
04015	478	40	**491**	**27**
04043	8,690	5,862	**9,608**	**4,944**
04048	419	387	**443**	**363**
04126	3,082	204	**3,154**	**132**
04746	30,731	**137**	**30,624**	244
04908	5,443	359	**5,557**	**245**
04936	32,833	6,812	**33,725**	**5,920**
05091	0	133	0	133
05121	32,575	1,164	**33,563**	**176**
05261	655	268	**766**	**157**
06426	52,104	1,006	**52,633**	**477**
06453	126	313	**130**	**309**
06995	27,072	448	**27,240**	**280**
07162	39,297	0	39,297	0
07859	61,891	0	61,891	0
07879	**39,944**	89	39,939	**49**
07910	**6,499**	**266**	6,440	325
08215	**32,958**	**170**	32,912	216
08219	12,627	1,528	**13,420**	**735**
08378	**10,995**	**478**	10,969	504
08405	45,005	88	**45,041**	**52**
08434	**2,307**	**0**	2,301	6
08455	44,103	159	**44,111**	**151**

Table 6.3: Comparison between the proposed classifier on the record 05091.

Approach	**TP**	**TN**	**FP**	**FN**
Zhou (2015) *et al.* [249]	0	36,644	0	133
MORPHYTHM — Random Forest	25	36,633	11	108
MORPHYTHM — J48	19	36,592	52	114
MORPHYTHM — Logistic	0	36,640	4	133
MORPHYTHM — AdaBoost M1	0	36,644	0	133
MORPHYTHM — RepTree	14	36,620	24	119

increases. Thus, we would like to study this specific scenario and observe if the application of a cost-benefit analysis can suite some specific constraints.

6.2.3 From Global to Local Predictions

This work has been motivated by the consideration that a local prediction of Atrial Fibrillation episodes can lead to better results if compared to a global one. Indeed, a local prediction strategy has been adopted in order to best contribute to the prediction of AF episodes and to evaluate if a local approach may be preferred instead of a global one. Thus, we present an extension of *MORPHYTHM* aiming at further improving its accuracy. We first performed a rigorous feature engineering process in order to identify the features that contribute the most to the prediction of AF events. Then, we experimented most advanced machine learning techniques, including artificial neural network and deep learning techniques. Finally, we integrated in *MORPHYTHM* the concept of "local" prediction, successfully used in another context [155]. Especially, instead of producing a single prediction model, the new version of *MORPHYTHM*, called *LOCAL MORPHYTHM*, automatically build several prediction models based on the characteristics of the ECGs in the training set. In particular, the training set is clustered in order to put together ECGs that exhibits similar characteristics. Then, for each cluster, *LOCAL MORPHYTHM* builds a prediction model. When a new data point is provided, *LOCAL MORPHYTHM* first selects the most suitable model based on the characteristics of the new data point, and then it performs the prediction applying the selected model.

Similarly to *MORPHYTHM*, also in *LOCAL MORPHYTHM* we consider both rhythm and morphological features. Especially, we consider the same set of features used in *MORPHYTHM* [139]:

The main difference between *LOCAL MORPHYTHM* and *MORPHYTHM* regards the way as the prediction is performed. In *MORPHYTHM*, as in any canonical approach based on supervised machine learning techniques, a training set is used to build a (global) prediction model. Such a model is used on all the new data points where a prediction is required. Especially, when a new heartbeat signal is provided, *MORPHYTHM* first computes the features on this new heartbeat signal and then uses the prediction model to determine whether or not the heartbeat is fibrillating or not fibrillating.

However, the heartbeat signals in the training set could be quite different each other. The heterogeneity of the training set might negatively impact the accuracy of the prediction model [155]. In order to mitigate such a problem, in *LOCAL MORPHYTHM* we integrated a local prediction strategy [155].

LOCAL MORPHYTHM first clusters the training set into homogeneous sets of heartbeat signals. Then, it builds for each cluster a specific prediction model using a supervised machine learning technique. In this way, *LOCAL MORPHYTHM* does not have just one global prediction model, but it has a set of prediction models that are particularly suitable for specific heartbeat signals.

When a new heartbeat signal is provided, *LOCAL MORPHYTHM* first computes the features on this new heartbeat signal and then it identifies the cluster of heartbeat signals more similar to the new heartbeat signal. Once identified such a cluster, *LOCAL MORPHYTHM* uses the model associated to the identified cluster of heartbeat signals to predict whether or not the new heartbeat is fibrillating or not fibrillating. The workflow of *LOCAL MORPHYTHM* is depicted in Figure 6.4.

Empirical Evaluation

This section reports the empirical evaluation we conducted to evaluate the accuracy of *LOCAL MORPHYTHM*.

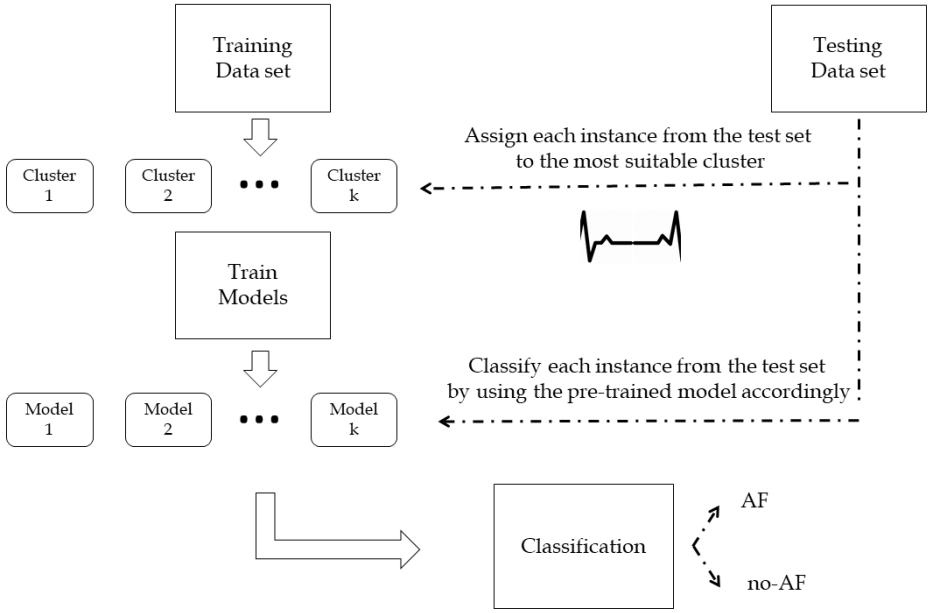

Figure 6.4: Workflow of *LOCAL MORPHYTHM*.

Designing a refined training In order to cluster the training set, we have exploited the k-means clustering algorithm [151]. This method follows a simple way to classify a given data set through a certain number of clusters fixed *a priori*. The main idea is to define k centroids, one for each cluster. The main steps are described below:

- Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
- Assign each object to the group that has the closest centroid.
- When all objects have been assigned, recalculate the positions of the K centroids.
- Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

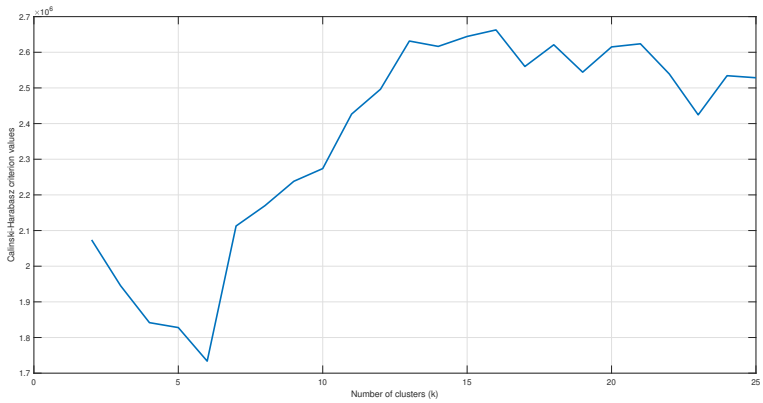

Figure 6.5: Results of the Calinski-Harabasz score to determine the best value of k for the k-means clustering algorithm. The higher the value of the score the higher the overall quality of the clustering.

We have determined the optimal value of k using the Variance Ratio Criterion (also known as Calinski-Harabasz score) [29]. Especially, we have have performed the clustering of the heartbeats by using different values of k from 1 to 25. For each cluster we have computed the Calinski-Harabasz score in order to determine the value of k that determines the clustering with the highest score. The plot in Figure 6.5 shows that the highest Calinski-Harabasz value occurs with $k = 16$. This number has also been confirmed by involving the Silhouette method [198], an alternative method for the identification of the best k value.

Features Selection For each heartbeat signal we extract a total number of 76 different features (eight rhythm features and 68 morphological features). In order to select the most appropriate features for the detection of AF events we used the Weka *InfoGainAttributeEval* as Attribute Evaluator and *Ranker* as Search Method. The former basically evaluates the worth of an attribute by measuring the information gain with respect to the class, while the latter ranks attributes by their individuals evaluations.

Table 6.4: Features ranking using Information Gain.

Rank	**InfoGain**	**Attribute**	**Type**
1	0.86	Entropy from Zhou (2015) *et al.* [249]	Rhythmic
2	0.20	Entropy from the rhythmic pattern	Rhythmic
3	0.18	heartbeat absolute length	Rhythmic
4	0.14	coeff. no. 10 from AR model	Morphological
5	0.13	coeff. no. 11 from AR model	Morphological
6	0.12	coeff. no. 7 from AR model	Morphological
7	0.11	coeff. no. 12 from AR model	Morphological
8	0.11	coeff. no. 1 from AR model	Morphological
9	0.11	coeff. no. 8 from AR model	Morphological
10	0.11	coeff. no. 9 from AR model	Morphological
11	0.11	coeff. no. 3 from AR model	Morphological
12	0.10	coeff. no. 6 from AR model	Morphological
13	0.10	coeff. no. 2 from AR model	Morphological
14	0.10	coeff. no. 4 from AR model	Morphological
15	0.10	coeff. no. 3 from FFT model	Morphological
16	0.10	coeff. no. 31 from FFT model	Morphological
...	...	...	...
75	0.03	Entropy of Sample Amplitudes	Morphological
76	0.01	Length discrete class	Rhythmic

The feature selection process has been conducted on the MIT-BIH AF Database [83]. The outcome of the features selection process is reported in Table 6.4. From the analysis of the results achieved, we observe that:

- rhythmic information in AF episodes detection represent the main contribution in terms of information gain;
- morphological features of an ECG can provide a contribution in terms of information gain for the automatic classification of heartbeats. Specifically, these features refer to the middle and the last part of the signal, where the fibrillating rhythm appears and where the P-wave can exhibit its changes.

By selecting a fixed threshold of 0.12, we obtain a selection of a group of six features containing a balanced number of morphological and rhythmic features. Thus, we decided to incorporate in *LOCAL MORPHYTHM* the first six features reported in Table 6.4.

Design of the Study The goal of this study is to evaluate the accuracy of *LOCAL MORPHYTHM* in classifying AF events in a patient. The perspective is both (i) of a researcher who wants to understand if a local prediction strategy to combine rhythmic and morphological features is worthwhile for detecting AF events, and (ii) of a practitioner who wants to use the most accurate and precise approach in a telemedicine application for the detection of AF events. Thus, the study is steered by the following research question:

To what extent, a local prediction model—based on the combination of rhythmic and morphological information—improves the automatic detection of AF episodes?

The context of this study is represented by the MIT-BIH AF Database [83], and specifically the $AFDB_2$, *i.e.*, the AFDB without records 00735 and 03665 because, for such records, only information on the rhythm is available [83]. Also, records 04936 and 05091 were excluded due to many incorrect manual AF annotations [144].

In the context of our study, we also experimented a large set of machine learning techniques. Indeed, for the classification performances, we have involved in our experiments—beyond the Random Forest [16], J48 [190], Logistic [50], AdaBoost M1 [69] and RepTree [60] already used by Laudato *et al.* to evaluate *MORPHYTHM* [139]—Neural Network [135], Multi Layer Perceptron [176], JRip [43] and SGD (which implements stochastic gradient descent for learning various linear models)[3].

As validation technique, we have chosen the L1PO-CV.

Analysis of the Results

Table 6.5 compares the prediction accuracy, in terms of TP, TN, FP, and FN, achieved by *LOCAL MORPHYTHM*, *MORPHYTHM*, and the approach proposed by Zhou (2015) *et al.* [249], the most accurate approach in the literature for the detection of AF events.

[3]https://weka.sourceforge.io/doc.stable-3-8/weka/classifiers/functions/SGD.html

Table 6.5: Comparison of *LOCAL MORPHYTHM* with *MORPHYTHM* (with the same features selection strategy used in *LOCAL MORPHYTHM*) and the approach proposed by Zhou (2015) *et al.* [249]. In boldface the best results achieved by these methods.

Approach	**TP**	**TN**	**FP**	**FN**
Zhou (2015) *et al.* on $AFDB_2$	457,001	**554,247**	**15,513**	12,966
MORPHYTHM — Random Forest	459,211	534,822	34,489	11,205
LOCAL MORPHYTHM — Random Forest	458,980	534,824	34,501	11,422
MORPHYTHM — J48	449,471	512,763	54,209	23,284
LOCAL MORPHYTHM — J48	446,947	513,453	55,259	24,068
MORPHYTHM — Logistic	463,730	545,621	22,184	8,192
LOCAL MORPHYTHM — Logistic	464,623	545,624	22,003	7,477
MORPHYTHM — AdaBoost M1	461,635	549,572	16,188	12,332
LOCAL MORPHYTHM — AdaBoost M1	461,214	547,589	18,287	12,637
MORPHYTHM — RepTree	451,962	522,829	42,931	22,005
LOCAL MORPHYTHM — RepTree	452,231	522,819	42,899	21,778
MORPHYTHM — 3-layers LSTM NN	462,730	545,621	22,484	8,892
LOCAL MORPHYTHM — 3-layers LSTM NN	460,076	546,799	23,081	9,771
MORPHYTHM — 3-layers Conv. NN	461,319	546,032	23,260	9,116
LOCAL MORPHYTHM — 3-layers Conv. NN	459,660	546,020	23,695	10,352
MORPHYTHM — MultiLayer Perceptron	457,595	544,031	26,964	11,137
LOCAL MORPHYTHM — MultiLayer Perceptron	457,606	544,017	26,992	11,112
MORPHYTHM — JRip	452,966	522,840	42,121	21,800
LOCAL MORPHYTHM — JRip	451,599	523,296	42,571	22,261
MORPHYTHM — SGD	464,227	545,921	21,577	8,002
LOCAL MORPHYTHM — SGD	**465,341**	545,388	21,565	**7,433**

From the analysis of the results emerges that for both the approaches *MORPHYTHM* and *LOCAL MORPHYTHM* the best overall accuracy is achieved when SGD is used as machine learning techniques.

Using such a technique, *LOCAL MORPHYTHM* is able to achieve the best results in terms of both TP and FN. Specifically, *LOCAL MORPHYTHM* is able to identify 8,340 TP more than the baseline (approach by Zhou (2015) *et al.*) and 1,114 TP more than *MORPHYTHM*. Also, *LOCAL MORPHYTHM* is able to retrieve less FN with respect to both the baseline and *MORPHYTHM*, *i.e.,*-5,533 and -569, respectively.

However, the approach proposed by Zhou (2015) *et al.* [249] is still the best in terms of TN and FP. Specifically, *LOCAL MORPHYTHM* and *MORPHYTHM* generate 6,052 and 6,064 FP more than the approach by Zhou, respectively. In

Table 6.6: Example of records on which *LOCAL MORPHYTHM* outperforms both *MORPHYTHM* and the approach by Zhou (2015) *et al.* [249] in terms of all the considered evaluation metrics.

Record	Interval	TP	TN	FP	FN
04043	Zhou (2015) *et al.* [249]	8,690	44,299	3,063	5,862
	Best *MORPHYTHM* – Logistic	9,608	43,565	3,797	4,944
	LOCAL MORPHYTHM – AdaBoost M1	**10,090**	**44,991**	**2,371**	**4,462**
06426	Zhou (2015) *et al.* [249]	52,104	815	1,229	1,006
	Best *MORPHYTHM* – Logistic	52,633	629	1,415	477
	LOCAL MORPHYTHM – SGD	**52,576**	**901**	**1,143**	**534**

terms of TN, instead *LOCAL MORPHYTHM* and *MORPHYTHM* retrieves less TN as compared to the baseline, *i.e.,*, -8,859 and -8,326, respectively.

By looking at the results achieved at patient level, *i.e.,* by considering a single recording, we observe that *LOCAL MORPHYTHM* sensibly outperforms—in terms of every metrics—both the baseline and *MORPHYTHM* for 5 out of 21 recordings (around 24%). Examples of such an improvement is reported in Table 6.6, where it is possible to observe the classification performances of *LOCAL MORPHYTHM* with respect to the baseline and *MORPHYTHM*.

In addition, if we focus the attention on just TP and FN, *LOCAL MORPHYTHM* outperforms both the other approaches baselines in 8 out of 21 recordings (around 38% of the data set).

For the remaining recordings, the value of all the evaluation metrics are almost balanced, in the sense that no significant improvement can be observed.

The only recording with abnormal classification performances is the recording 08378 where *LOCAL MORPHYTHM* presents a significant loss in terms of TP and FN with respect to the other two approaches. This suggests that on this particular recording the local prediction strategy is not worthwhile because very likely such a recording exhibits characteristics that are quite different from the other recordings in the data set.

In order to validate such a conjecture, we compare the average distance between each recording and all the others but 08378 and the distance between each recording and recording 08378.To compute the distance between two recordings we considered them as mono dimensional vectors (by selecting the first ECG

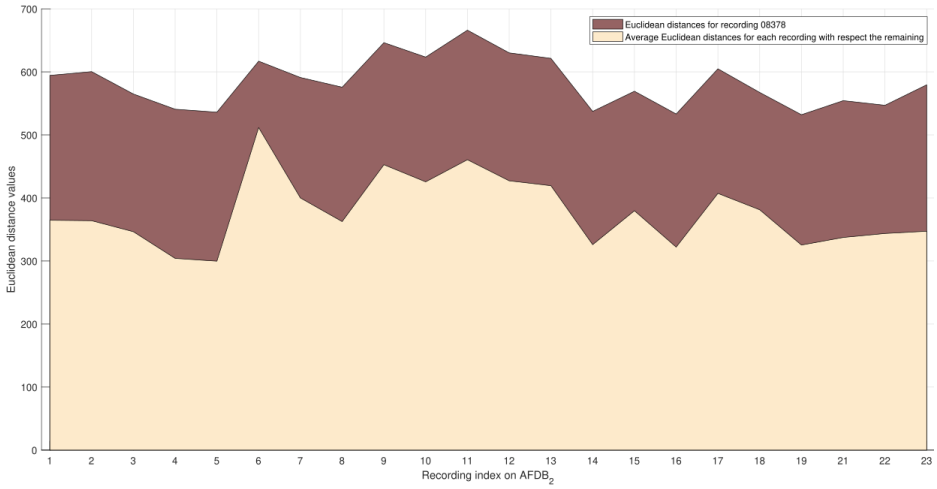

Figure 6.6: Average distance between a generic recording i and all the other recordings but 08378 compared to the distance between recording i and 08378.

channel available for each recording) and then compute the Euclidean distance between the two vectors.

The analysis is depicted in Figure 6.6. As we can see, the distance between the recording 08378 and a generic recording i is much higher that the average distance between the recording i and all the other recordings but 08378. Such a result confirms our conjecture that the recording 08378 is quite different from the others recording; thus, in this specific case, the local prediction strategy does not provide any benefits as compared to the other two approaches.

Once this recording is excluded from the data set, the classification accuracy of *LOCAL MORPHYTHM* improves even more. Indeed, *LOCAL MORPHYTHM*—especially when using the Logistic and the SGD algorithms—avoids a loss of around 1,5 thousand heartbeats classified as TP and FN.

Final remarks

An experimentation conducted on the MIT-BIH AF Database [83] indicates that *LOCAL MORPHYTHM* can increase the TP and reduce the FN as compared to *MORPHYTHM* and the approach by Zhou (2015) *et al.* [249], one of

the best approaches in the literature for the detection of AF episodes. Future work will be devoted on the one hand on the replication of the experimentation on other data sets in order to corroborate the results achieved on the MIT-BIH AF Database and on the other hand on the application of a local prediction technique in the context of automatic detection of other types of arrhythmia.

6.2.4 Explainable Predictions for AF Detection

Supported by the results of these previous works, where a machine learning approach has been defined to combine morphological and rhythmic information to support the identification of AF events, we conjecture that there is still room for improvement. Indeed, despite the promising results achieved, the main limitation of these approaches is represented by the difficulties to explain a specific prediction. Indeed, all the features extracted from the ECG are put together in one single learning algorithm. This makes difficult to identify the event that triggered the prediction. Based on the willingness of having an accurate and explainable method for detecting AF events, we designed *AMELIA* (AutoMatic dEtection of atriaL fIbrillation for heAlthcare). The proposed approach aims at simulating as much as possible the doctor behaviour during the detection of AF episodes. Especially, *AMELIA* first analyze the morphology of the heartbeat in order to identify the absence of p-wave and then confirm the anomaly by checking the presence of arrhythmia. We believe that *AMELIA* can be better employed in tele-medicine applications, where e-AI (explainable-Artificial Intelligence) is often a strong requirements. Indeed, *AMELIA* – thanks to the conceptual and *de facto* separation of the data sources between rhythmic and morphological – can provide highly accurate information in the process of diagnosis definition (which will be submitted to a medical doctor). For example, *AMELIA* – beyond the generation of a warning indicating a potential AF episode – can provide additional information, such as the heartbeats not showing a P wave or the ECG segment with an irregular rhythm.

An AF episode is diagnosed by a doctor when the morphology of the heartbeat is abnormal and there is a simultaneous arrhythmia. *AMELIA* aims at simulating as much as possible such a behaviour. The workflow of *AMELIA* is depicted in Figure 6.7. In the preprocessing stage of the raw ECG, *AMELIA* extracts all the

heartbeat signals and all the R peak positions. These signals are submitted to a *Morphology analyser*.

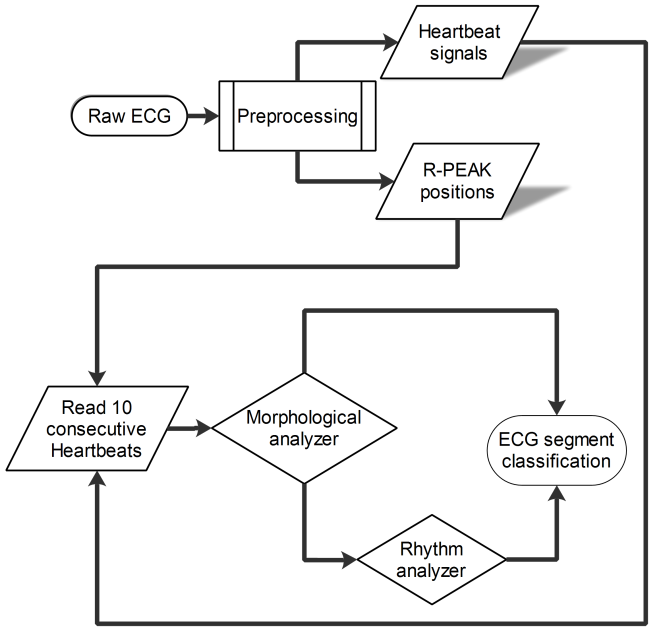

Figure 6.7: *AMELIA* workflow

If the morphology of the heartbeat is abnormal, the *Morphology analyser* triggers the *Rhythm analyser*. The *Rhythm analyser* takes as an input the extracted R peak positions and tries to consolidate the initial warning identified by the *Morphology analyser*. If the *Rhythm analyser* identifies through the analysis of ten consecutive R-R intervals an arrhythmia, then an *AF episode* is identified. Otherwise, the initial warning of the *Morphology analyser* is rejected. In this case the abnormal morphology of the heartbeat could be due to a wrong classification of the NN or just to some noise in the ECG. In the following subsections we provide more details on each component of *AMELIA*.

Definition of a heartbeat It is necessary to clarify the concept of *heartbeat signal*. In *AMELIA*, a heartbeat signal is a raw ECG segment included between two consecutive R peaks (see Figure 6.8).

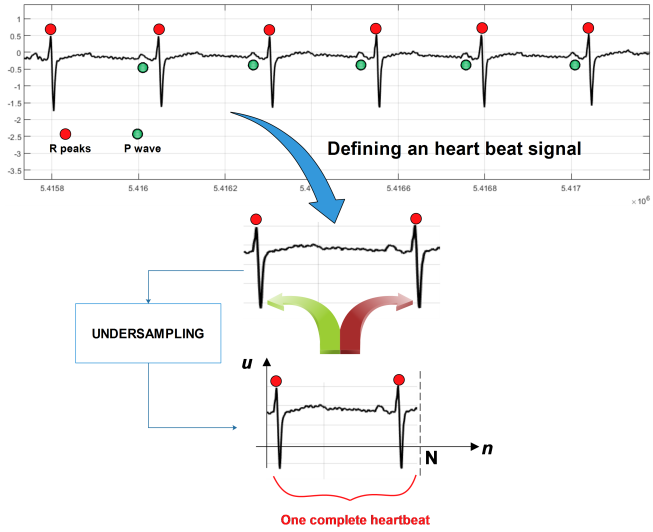

Figure 6.8: Definition of heartbeat signal in *AMELIA*

The choice to define a heartbeat signal in this way is due to the consideration that the morphological features – observable during AF episodes – are (i) the absence of *P wave* and (ii) the potential fibrillation waves in its place. The concept of *heartbeat signal* is also faced in the work by Xu *et al.* [241] with the difference that the authors define it as the signal between the two middle points of three consecutive R peaks. We decided to work with heart dynamics included between two consecutive R peaks because the fibrillating phenomena are inscribed between those points. We used the Pan-Tompkins method [179] to obtain all the expected heartbeat signals of a given full ECG signal.

A complete heartbeat is represented by a vector defined as:

$$hbs = u_1, u_2, ..., u_N \tag{6.1}$$

where u_1 and u_N are the raw amplitudes of the samples corresponding to the position of the left and right R peak respectively (see Figure 6.9).

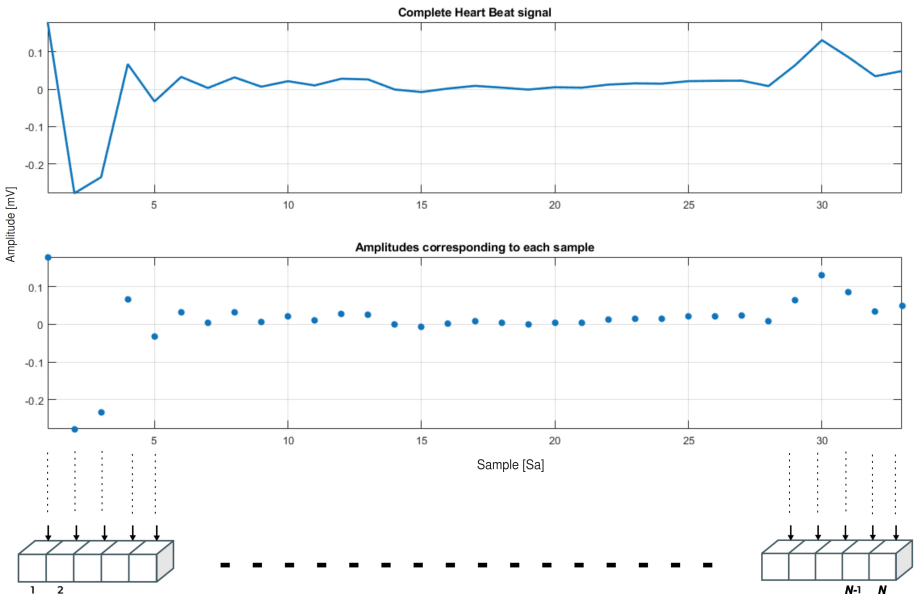

Figure 6.9: Representation of a complete heartbeat in *AMELIA*.

It is worth noting that – to provide fixed-length instances to the *Morphology analyser* component – all the heartbeat signals have been submitted to a process of down-sampling. N is the fixed length of each complete heartbeat signal.

Morphology Analyser The *Morphology Analyser* is in charge to analyse the morphology of a heartbeat. The input of this component is represented by a heartbeat. Ideally, the output is **no-AF** if the morphology of the heartbeat is normal and **AF** if the morphology of the heartbeat does not have the P wave and shows fluctuating waveforms (f-waves), *i.e.,* the morphological characteristics of a heartbeat in presence of an AF event. The morphology classification of the heartbeat is based on a Recurrent Neural Network (RNN) [102] with multiple LSTM cells. The choice is justified by the consideration that LSTM cells better adapt to time-series classification [117] (as in the case of ECG).

Rhythm Analyser The *Rhythm Analyser* aims at identifying normal rhythm or arrhythmia by evaluating a buffer of ten consecutive beats. It is worth noting that,– based on a consolidated opinion from cardiologists – ten consecutive heartbeats can be retained enough to diagnose atrial fibrillation. This number is also confirmed by the works in [130, 255] where a minimum of 3 and 6 consecutive heartbeats has been evaluated.

In details, the *Rhythm Analyser* classifies each beat as *short*, *long*, or *normal*. Considering that the normal heart rate during rest for teenagers is around 70-120 beat per minute (bpm) and adults is around 60-90 bpm [53], each beat is classified as follows: ***short*** if bpm $>$ 120, ***long*** if bpm $<$ 50 and ***normal*** otherwise. Once the *Rhythm Analyser* has buffered and labelled ten consecutive heartbeats, computes the entropy of the buffer B. Entropy is a logarithmic measure of the number of states with significant probability of being occupied:

$$H(B) = - \sum_{i \in \{short, long, normal\}} p_i \cdot log(p_i)$$

where p_i is the probability that the beat belongs to the i^{th} category of beat. This means that the higher the entropy of the buffer of heartbeats, the higher the likelihood that the rhythm is unstable, indicating the presence of arrhythmia. When the entropy is higher than a pre-specified threshold δ, the rhythm is considered abnormal.

Putting All Together Algorithm 1 shows how the *Morphology Analyser* and the *Rhythm Analyser* are combined in order to detect AF events.

Algorithm 1 Detection of Arrhythmia

```
Require: ECG                                                    ▷ Raw ECG
  HBS = ExtractHeartBeatSignals(ECG)
  RRI = ExtractRRInterval(ECG)
  for each hbs_i ∈ HBS do
    Morphology = MorphologyAnalyser(hbs_i)
    if Morphology == AF then
      buffer_i ← ∅                          ▷ new buffer for the i^th heartbeat
      BUFFERS ← BUFFERS ∪ buffer_i
    end if
    for each buffer_j ∈ BUFFERS do
      buffer_j ← buffer_j ∪ RRI_i
      if size(buffer_j) == MAX_SIZE then
        Rhythm = RhythmAnalyser(buffer_j)
        if Rhythm == ABNORMAL then
          GenerateWarning()
        end if
        BUFFERS ← BUFFERS \ buffer_j
      end if
    end for
  end for
```

For each heartbeat signal, a fixed length buffer hbs_i is instantiated, containing the amplitudes of the signal. The buffer hbs_i then is submitted to the *Morphology Analyser*, which provides its classification. When the morphology is classified as *AF*, a new buffer of heartbeats is created. Once the buffer of heartbeats has reached the max size (set as 10, in our case), it is submitted to the *Rhythm Analyser*. Based on the entropy information evaluated on the buffer, a classification in terms of rhythm is provided. If also the rhythm is identified as *IRREGULAR*, a warning is generated.

Empirical Evaluation

We compared *AMELIA* to the method proposed in [249], where AF episodes are identified by using only a RRI analysis. Thus, in the context of the study we formulated the following research question:

Does AMELIA outperform state of the art AF detection approaches?

We chose as baseline the approach by Zhou (2015) *et al.* [249] because in the state of the art it is one of the most accurate approaches based on RRI analysis. We also keep *Morphythm* [138] as a reference, because it is based on a combination of RRI and morphological analysis, too.

Context of the Study The proposed approach has been experimented on the MIT-BIH AFDB[82].

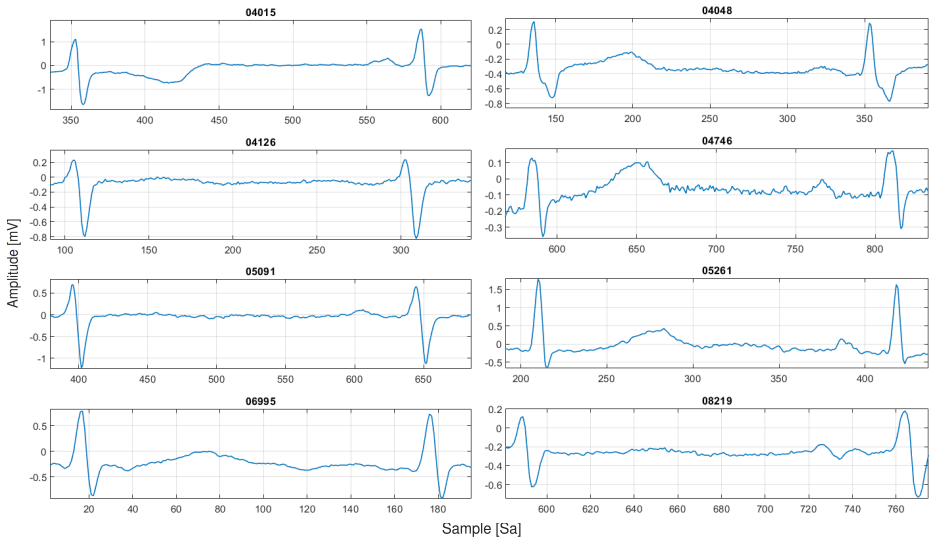

Figure 6.10: The records chosen for this study.

The other records have been ignored. Examples of ignored records are #06453 and #08455 where, in the available signal 1 from the database, the shapes differ from the ones of the above group of recordings (Figure 6.11).

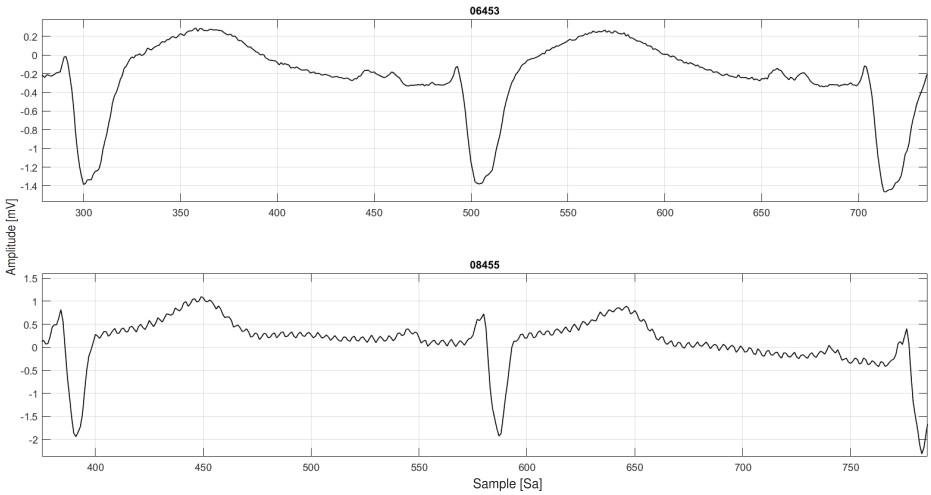

Figure 6.11: The records ignored for the study.

With the R-peak positioning labels provided by Physionet, each heartbeat signal has been manually observed and analysed, with the help of a medical team. Only signals presenting a clear AF effect have been selected. The operation has been carried out for all the chosen records. A total of 1637 heartbeat signals from the 8 different recordings have been manually extracted. The minimum length for each of these signals varies from 33 to 111 samples. By doing that, we have obtained two types of signal: *AF* and *Normal (no-AF)* heart-beat signals.

Training of the LSTM RNN The training of the LSTM RNN has been performed on a balanced data set composed of 1637 instances for the **AF Class** and 1653 for the **no-AF Class**. To the aim of guaranteeing an alignment, all the instances have been downsampled to 33 points. An example of selected and downsampled instances is depicted in Figure 6.12.

The LSTM parameters have been experimentally defined, through a *trial & error* approach. To validate the network, a classical L1PO cross validation has been applied to the data set.

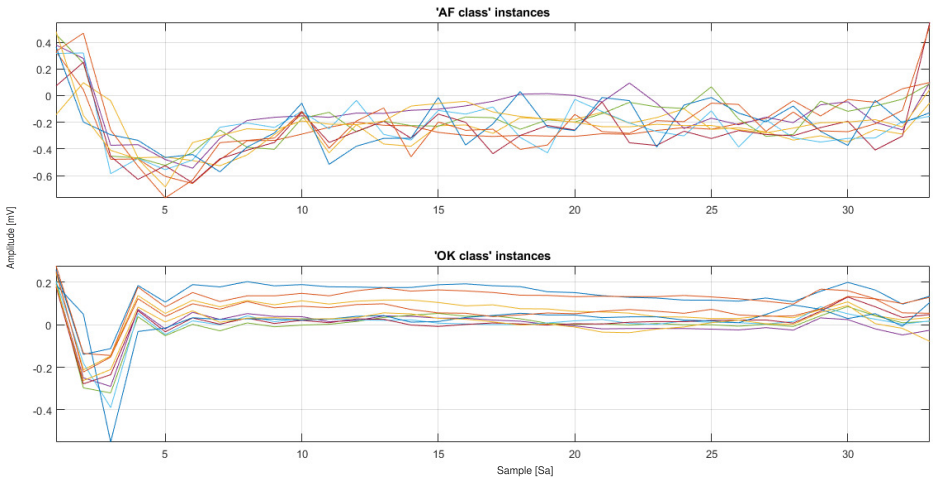

Figure 6.12: Examples of the manually selected instances from the Physionet AFDB

Clustering Before reporting the classification results, we aim at validating the manual selections performed in the previous steps. To this aim, we have selected one AF-labelled heartbeat from each of the chosen records and we have described each of these heartbeats using several descriptors, as described below:

- **Entropy**
  For each heartbeat signal, we have evaluated 3 types of entropy. The first one is the entropy according to the method proposed in [249], the second one is the novel operator previously reported, while the morphological entropy is the Shannon entropy evaluated on the amplitude samples of a heartbeat signal.

- **Classical Statistical Features**
  We have used also some common descriptors, such as: the absolute length of the heartbeat, the heartbeat classification, the mean, variance and norm of the amplitude samples. For the sake of accuracy, we have also proportionally divided each heartbeat in 10 segments, and we have evaluated the mean and variance of the amplitude samples for each obtained segment.

- **Fast Fourier Transform**
  For each heartbeat signal, we have applied the Fast Fourier Transform operator on 128 points.

- **AR model coefficients**
  For each heartbeat signal, we have applied an AR model of order 4 on 128 points.

After creating the data set of indicators, we have applied a technique of unsupervised clustering. We have chosen the Hierarchical Clustering, with *Euclidean distance* as the similarity function and the *average* as the agglomeration method. We have obtained the dendrogram depicted in Fig 6.13. Thus, even if they seemed to have a common ECG waveform shape – when they are observed from an AF heartbeat perspective – the clustering process assigns them to distinct groups. Of course, the height around 30 in the dendrogram would produce one single – not fine – cluster for our records.

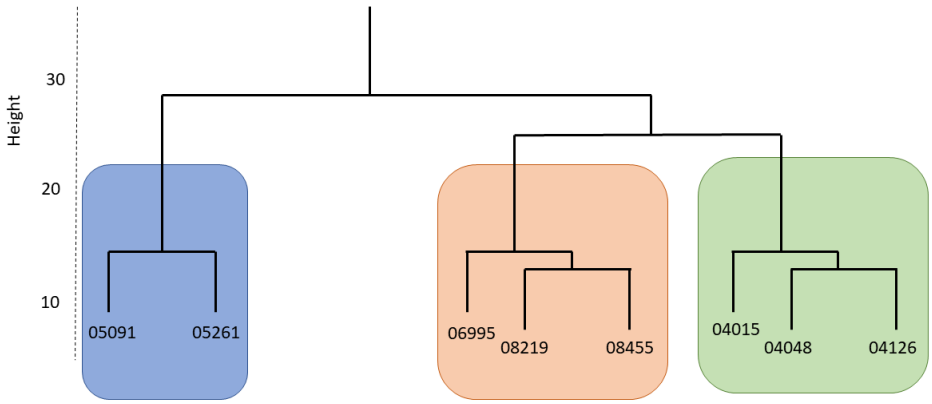

Figure 6.13: The dendrogram for the manually selected records from AF Database, based on a AF heartbeat

From this step of clustering, we obtained a refined view of the groups to which our records belong.

Table 6.7: *AMELIA* classification performance compared to the chosen baseline on MIT-BIH AF-db cluster 1

Cluster 1	**Method**	**TP**	**TN**	**FP**	**FN**
04015	*AMELIA*	500	42088	2886	25
	Morphythm	491	40650	2836	27
	Zhou	478	40707	2779	40
04048	*AMELIA*	792	38967	165	90
	Morphythm	443	38982	145	363
	Zhou	419	38990	137	387
04126	*AMELIA*	3345	39581	960	51
	Morphythm	3154	38149	1424	132
	Zhou	3082	38743	830	204
Metrics			**Sens**	**Spec**	**Acc**
AMELIA			**0,965**	0,968	**0,968**
Morphythm			0,887	0,964	0,961
Zhou			0,863	**0,969**	0,965

Analysis of the Results

We have experimented the proposed approach on two freely accessible data set, the Physionet MIT-BIH AFBD and the Normal Sinus Rhythm Database (NSRDB) [82].

The classification results are shown in Tables 6.7, 6.8, 6.9 according to the clusters previously obtained. The difference in number of heartbeats is due to the nature of *AMELIA*. Our online tool embeds the Pan-Tompkins algorithm as peak detector. Even if highly accurate, the performances of *AMELIA* integrate an additive error due to potential wrong classifications of this algorithm. Therefore, the only way to compare *AMELIA* with respect to the chosen baseline, is to evaluate the Sensitivity, Specificity and Accuracy overall metrics.

From the achieved results, it is possible to observe that for records belonging to:

- **Cluster 1:**
  AMELIA outperforms both the baseline and *Morphythm* by keeping a comparable specificity and better sensitivity and accuracy;

Table 6.8: *AMELIA* classification performance compared to the chosen baseline on MIT-BIH AF-db cluster 2

Cluster 3	**Method**	**TP**	**TN**	**FP**	**FN**
05091	*AMELIA*	44	35470	986	98
	Morphythm	0	36640	4	133
	Zhou	0	36644	0	133
05261	*AMELIA*	881	43739	1953	51
	Morphythm	766	43015	1595	157
	Zhou	655	44215	395	268
Metrics			**Sens**	**Spec**	**Acc**
AMELIA			**0,861**	0,964	0,963
Morphythm			0,725	0,980	0,977
Zhou			0,620	**0,995**	**0,990**

Table 6.9: *AMELIA* classification performance compared to the chosen baseline on MIT-BIH AF-db cluster 3

Cluster 3	**Method**	**TP**	**TN**	**FP**	**FN**
06995	*AMELIA*	11215	27160	490	17784
	Morphythm	27240	25901	1767	280
	Zhou	27072	25648	2020	448
08219	*AMELIA*	7946	42595	4286	6207
	Morphythm	13420	40934	4203	735
	Zhou	12627	42637	2500	1528
08455	*AMELIA*	32705	15265	22	12470
	Morphythm	44111	15244	45	151
	Zhou	44103	15250	39	159
Metrics			**Sens**	**Spec**	**Acc**
AMELIA			0,587	0,947	0,768
Morphythm			**0,986**	0,932	0,959
Zhou			0,975	**0,948**	**0,962**

- **Cluster 2:**
  AMELIA provides significantly higher sensitivity with slightly lower specificity and accuracy;

- **Cluster 3:**
  AMELIA presents a significant loss, mostly in terms of sensitivity and accuracy.

From the L1PO-CV, we have chosen the best network in terms of accuracy on the test data set. With this, we have been able to experiment the proposed approach also on the MIT-BIH Normal Sinus Rhythm Databases. In this DB, all the recordings present a shape with high similarity with respect to the ones used in this study. The individuals included in the NSRDB were found to have had no significant arrhythmia. Tables 6.10 shows the results achieved on this DB. This experimentation represents a boundary validation for our proposed methods, because the goal is to avoid all the phenomena with (different) arrhythmia by using the useful information provided by the morphological module of *AMELIA*.

In this validation of *AMELIA*, we expected that the proposed tool does not get confused with (not significant) arrhythmia episodes affecting the patient from this data set. As we can see from the tables - by using our rhythm analyser - an arrhythmia can be detected as an Atrial Fibrillation episode. With the introduction of the morphological Analyser in *AMELIA* we have reduced the chance of misclassifying many heartbeats.

Final Remarks

Future works will be devoted to further improve the accuracy of *AMELIA*, by replacing the manual selection of ECG recordings with a fully automated process. With this step, we aim at training models based on very specific data. For example, the first level of Neural Network could provide improvements also for records with a particular shape (e.g. due to some hidden disease) or coming from a specific lead. We believe that this structure could help the classification also on the recordings belonging to cluster 2, due to the building of a morphology-specific classifier. We also plan to improve the classification performances, by refining some clue parameters in our proposed method. We specifically refer to

Table 6.10: *AMELIA* accuracy on MIT-BIH NSR-db

Records	***AMELIA* only rhythm**	**Full *AMELIA***
No.	# FP	# FP
16265	2	0
16272	0	0
16273	1	0
16420	1	0
16483	0	0
16539	13	0
16773	0	0
16786	0	0
16795	4	0
17052	29	0
17453	0	0
18177	5	0
18184	0	0
19088	10	1
19090	0	0
19093	0	0
19140	0	0
19830	16	14

the down-sampling resolution, the Neural Network structure, and the length of the rhythm pattern. Finally, the replication of the experimentation of *AMELIA* on other data sets is also part of our agenda for future work.

6.3 Detection of Arrhythmia conditions

In the context of an ECG analysis system, we focused on the detection of several arrhythmia conditions, such as Bundle Branch Block (BBB), Premature Ventricular Contractions (PVC) and Atrial Premature Beats (APB).

6.3.1 Motivations

Here we describe the motivations and incidences of the arrhythmia conditions of interest.

A bundle branch block can be defined as an abnormality of the electrical conduction system of the heart [66]. In case the defect is originated in the left or right ventricles the blocks are further classified into Right BBB (RBBB) and Left BBB (LBBB). Scientific research studies have reported that BBB has been observed in 8% to 18% of subjects with acute myocardial infarction. It has also been associated with an increased risk of complete heart block and sudden death [173, 127]. Before the involvement of thrombolytic treatment—that limits infarct size, improves ventricular morphology and function, and decreases mortality—several studies had reported on the incidence of RBBB in patients with acute myocardial infarction [154]. The range of incidence rate was found to be between the 3% and 29% [114, 45].

It was also found that RBBB is usually the manifestation of infarctions. These latter are often accompanied by heart failure, complete AV block, arrhythmias, and a high mortality rate [196, 8, 169]. With regard to the LBBB, the incidence in the general population is low, approximately 0.6% of subjects developing it over 40 years [41, 106]. The incidence rate changes if considering patients with chronic heart failure. Indeed, approximately one third of these patients have left bundle branch block (LBBB) on their 12-lead ECG [213, 10].

In the absence of structural heart disease, frequent PVCs have traditionally been considered a benign phenomenon, only requiring medical attention when symptomatic. This understanding has undergone a substantive evolution over the last decade. So-called benign PVCs are now known to have malignant potential in susceptible patients and can manifest as triggers for ventricular fibrillation (VF) and sudden cardiac death [107].

Ranging from 20% to 25% of ischemic strokes occur due to embolic complications caused by atrial fibrillation [64, 97]. In addition, for patients that have experienced ischemic stroke or transient ischemic attacks, in presence of AF they can be exposed to recurrent strokes [235]. Therefore, it is vital to detect paroxysmal atrial fibrillation after stroke or transient ischemic attack and involve anticoagulation treatment in such patients [98, 230]. This diagnose typically includes

a 24 hours continuously monitoring. One of the clues that can lead to a early diagnosis of paroxysmal atrial fibrillation are the occurrence of atrial premature beats (APB). Indeed, in 24-hour ECG recordings frequent APB are correlated to an increased incidence of paroxysmal AF in patients with ischemic stroke [234].

6.3.2 Real-Time Beat-to-Beat Arrhythmia Detection

The combination of several real-time features from the scientific literature has motivated this work. Indeed, we aimed at defining a novel method for the real-time classification of arrhythmia conditions. Therefore, we devised *NEAPOLIS* (**N**ov**E**l **AP**proach for the aut**O**matic rea**L**-time beat-to-beat detect**I**on of arrhythmia condition**S**) an approach that performs the classification of heart beat by extracting a set of features from an ECG trace and providing them to a machine learning component. The common characteristic among all the features that *NEAPOLIS* extracts from the ECG is that they are real-time, *i.e.,* they do not need any long-term observation of the ECG.

NEAPOLIS is an online detector of important arrhythmia conditions, such as BBB and PVC, based on the analysis of heartbeat signals. The high-level workflow of *NEAPOLIS* is depicted in Figure 6.14.

Once buffered a small segment—*i.e.,* at least 11 heartbeats—of a single lead digital ECG signal, *NEAPOLIS* operates to compute a beat-to-beat segmentation. Then, a 2-step median filter is applied to get rid of baseline drifts. Finally, *NEAPOLIS*—through specific algorithms—evaluates the features on the signal, scale them and creates the final feature vector to be submitted to the machine learning model as input. Last task of *NEAPOLIS* is to provide a label for the most probable classification among *N* (Normal Sinus Rhythm), *RBBB* (Right Bundle Branch Block), *LBBB* (Left Bundle Branch Block), *PVC* (Premature Ventricular Contraction), and *APB* (Atrial Premature Beat). Next subsections describe the main components of *NEAPOLIS* in detail. The accuracy of *NEAPOLIS* has been compared to the approach proposed by Pandey *et al.* [181] since—to the best of our knowledge—this approach is one of the most accurate in the literature and provides the same 5-class classification of heart beat of *NEAPOLIS*. However, the method proposed by Pandey *et al.* [181] requires a long-term observation of the ECG, by extracting from features the ECG trace that are computed

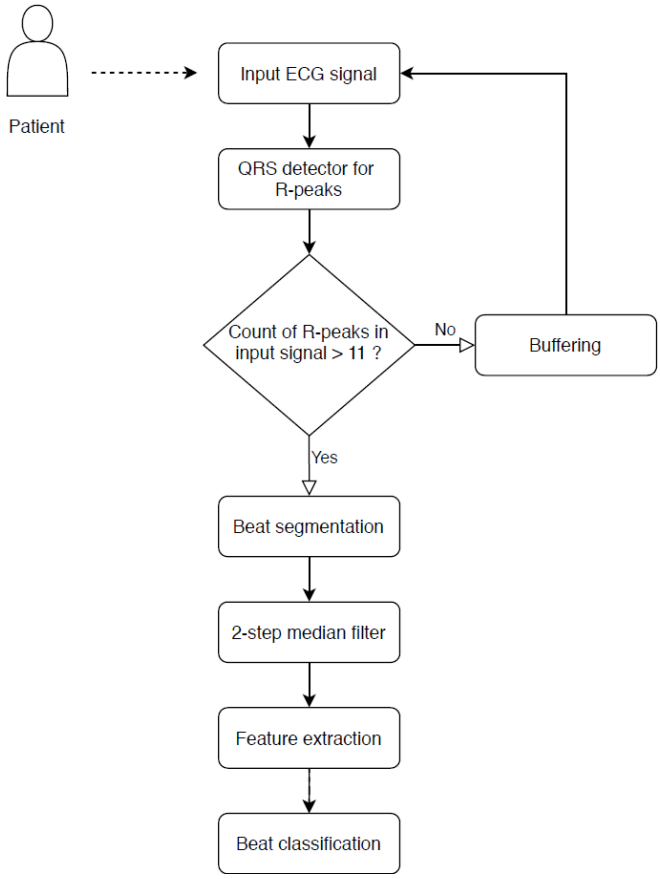

Figure 6.14: The workflow of *NEAPOLIS* for online beat classification.

on the past 20 minutes of the ECG. The unique characteristics of *NEAPOLIS* allows to obtain a classification in a much shorter time. Indeed, in *NEAPOLIS* eleven beats are required to compute all the features used by our approach to perform the classification. Therefore, for a subject with a heart rate value of 60 bpm the first classification can be performed after 11 seconds + $\mathbf{t}$, where $\mathbf{t}$ represents the computational time of *NEAPOLIS* to build and classify the features vector (that is, however, negligible). An empirical evaluation conducted on

the Physionet MIT-BIH arrhythmia database provides evidence of the benefits provided by *NEAPOLIS* also in terms of classification accuracy.

ECG Digital Processing The digital signal processing embedded in *NEAPOLIS* can be conceptually divided in preprocessing and main processing. Both these procedures are triggered only when a long enough portion of a digital single lead ECG is buffered. Once these two steps are completed, the features can be extracted from the obtained signal.

Preprocessing The preprocessing step of *NEAPOLIS* is the same proposed by Pandey *et al.* [181]. Therefore, only the baseline removal has been performed. Specifically, it concerns with the application of two median filters: a median filter of 200 ms is applied on the raw signal, a second median filter of 600ms is applied on the resulting signal from the previous step.

Beat-to-beat Segmentation This procedure is the same proposed by Pandey *et al.* [181]. Especially, *NEAPOLIS* needs to embed a QRS detector, such as the consolidated algorithm proposed in [180]. Once evaluated each R peak position in the buffered ECG, the segmentation process can start. The procedure is based on the evaluation of a window of 180 samples to be centered on an R peak. After, the selection of the samples included in the window is performed. This leads to the definition of a heartbeat signal, *i.e.,* a sample vector of length 180 centered on an R peak.

Heartbeat Features Due to their promising performance in prior similar works, we combined a set of morphological features already used in literature for ECG classification. *NEAPOLIS* differs from the state of the art approaches because of the constraint on the real-time detection. Indeed, only a very limited buffering of an ECG signal is needed so that the detection of arrhythmia is promptly offered. Next points describe in detail the features extracted by *NEAPOLIS*.

- **Energy of Maximal Overlap Discrete Wavelet Transform**:
  The wavelet transform (WT) is a mathematical operator that can be used

for the decomposition of time series signals into distinct subsignals. One of the two forms of WT is the DWT. The maximum overlap discrete wavelet transform (MODWT) is a modified DWT. In the MODWT, there is no process of subsampling, therefore leading to a higher level of information in the resulting wavelet and scaling coefficients, when compared to the DWT [78]. For our purposes, we evaluated the MODWT and then extracted the energy features according to the following steps: (i) selection of a mother wavelet function W and the decomposition level L; (ii) decomposition of the original heartbeat signals according to the specified W and L; and (iii) calculation of the energy of each coefficient in each node in the last level L. This procedure has also been partially considered in the feature extractor proposed by Li *et al.* [148]. In our case, we used *db2* as Daubechies wavelet function and three levels of decomposition.

- **Autoregressive Model (AR)**:
  As suggested in the method proposed by Zhao *et al.* zhao2005ecg, we involved the calculation of the Autoregressive model (AR) coefficients of order 4. As outcomes, we evaluated the AR coefficients and the reflection coefficients, using the Yule-Walker estimator [70].

- **Multifractal Wavelet Leader**:
  The goal of multifractal analysis is to study signals that present a pointwise Holder regularity variable, *i.e.,* that may largely vary from point to point. When dealing with a signal, performing the multifractal analysis refers to the estimation of its spectrum of singularities. Therefore, the determination of the spectrum of singularities of a signal is important to analyze its singularities [147]. In case of a real-life signal, it cannot be numerically evaluated due to constraint like finite resolution and the sampling of signals [137]. To overtake this limitation, a multifractal formalism was introduced: the wavelet leaders [109]. In *NEAPOLIS*, we involved the multifractal wavelet leader estimates of the log-cumulants of the scaling exponents.

- **Fast Fourier Transform**:
  Our approach embeds the evaluation of the Fast Fourier Transform on the

heartbeat signal. Indeed, FFT represents a method for extracting helpful information out of statistical features of ECG signal.

- **R-R interval descriptors**:
  This set of features is basically composed of three features:
  - *pre-RR interval*: the distance between the actual and previous heartbeat;
  - *post-RR interval*: the distance between the actual and next heartbeat;
  - *local-RR interval*: the average of 10 previous pre-RR values.

These features have been proposed by Pandey *et al.* [181], where they belonged to a larger set of R-R statistical descriptors. We opted to embed in *NEAPOLIS* only the features with an acceptable ECG buffering. Indeed, we avoid integrating in *NEAPOLIS* the *global-RR interval* presented by Pandey *et al.* [181] because it represented the average of all the pre-RR values present in the last 20 min. This would have compromised the constraint of *NEAPOLIS* to be a real-time detector.

Beat Classification Once extracted, the features described in Section 6.3.2 are normalized, to transform the features in a predefined range of values. We also apply a technique of sampling of the instances to deal with data unbalance.

After these further elaborations, the features are provided to a machine learning classifier for the final classification of the heartbeat in N (Normal Sinus Rhythm), $RBBB$ (Right Bundle Branch Block), $LBBB$ (Left Bundle Branch Block), PVC (Premature Ventricular Contraction), and APB (Atrial Premature Beat). *NEAPOLIS* has not been designed for a specific machine learning technique. The only constraint is represented by the use of a supervised technique. During the evaluation of *NEAPOLIS* we experimented several machine learning techniques.

Empirical Evaluation

The goals of this study are (i) understanding which are the most important descriptors of a heartbeat signal in applications of automatic detection of

arrhythmia conditions, such as the LBBB, RBBB, PVC and APB and (ii) comparing *NEAPOLIS* with the selected baseline. Thus, our study is steered by the following research questions:

> RQ_1: *What are the most important features for the beat-to-beat classification of arrhythmia conditions?*
>
> RQ_2: *Which is the accuracy of NEAPOLIS?*

With these research questions, we can distinguish two objectives. With RQ_1, we want to understand if some of the features we define can be discarded to obtain a higher classification accuracy while with RQ_2 we want to see if *NEAPOLIS* can reach a classification accuracy comparable to similar state of the art methods, especially to those that can be classified as off-line approaches, *i.e.,* that embed features requiring a long-term observation of an ECG.

Context of the Study The context of our study is represented by the Physionet MIT-BIH arrhythmia database [84, 160], a state-of-art database widely used in literature as reference data set for arrhythmia detection [160]. It is composed of 48 ambulatory ECG recordings. The acquisition was performed with a sampling frequency of 360 Hz. Each recording has two channels available: one is the modified lead II (MLII) and the other can vary between V1, V2, V4 or V5. Heartbeat annotations were provided by cardiologists. The total number of labelled heartbeats is approximately 110,000 divided into 15 different beat types.

According to a consolidated procedure on this database [240], the records with paced beats, namely 102, 104, 107 and 217 have been excluded from the study. The experiment was conducted on the remaining 44 records and considering 5 types of beats annotations: N, LBBB, RBBB, APB and PVC. Figure 6.15 shows the distribution of such types of beats in the dataset.

Experimental Procedure This section details the experimental procedure we follow to answer our research questions.

- RQ_1**: Feature Analysis**:
  Using *wfdb*[4] toolkit we extracted raw signals and annotations from the

[4]https://archive.physionet.org/physiotools/wfdb.shtml

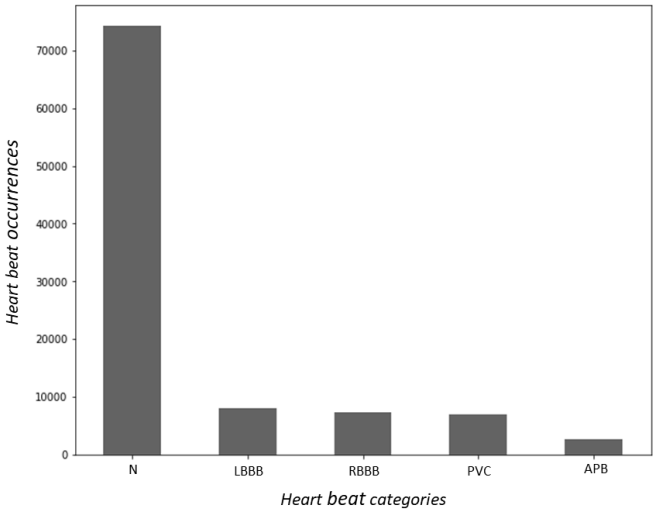

Figure 6.15: Count of selected heartbeat types from the MIT-BIH arrhythmia database [160].

arrhythmia database. Since the annotations contain both R-peak positions and beat types, we used the former information to split the signals in beat segments and the latter to filter beats by the selected types for this study. After this, we preprocessed the signals following the procedure detailed in Section 6.3.2. Finally, we subtracted the filtered signal from the raw one, obtaining a signal with corrected baseline, as depicted in Figure 6.16.

For each ECG segment obtained from the above elaboration steps, we computed the features generation through the algorithms previously described. The features vector was therefore composed of the record id (a code used by Physionet to indicate a patient), the computed features and the label indicating the heartbeat class.

To answer RQ_1, we conducted a features analysis on this data set. The first step has been focused on an analysis based on the Pearson correlation coefficient r. Indeed, we removed the features having r greater than 0.9. Afterwards, we did another step of features selection based on importance weights using a tree-based classifier as estimator. The features importance

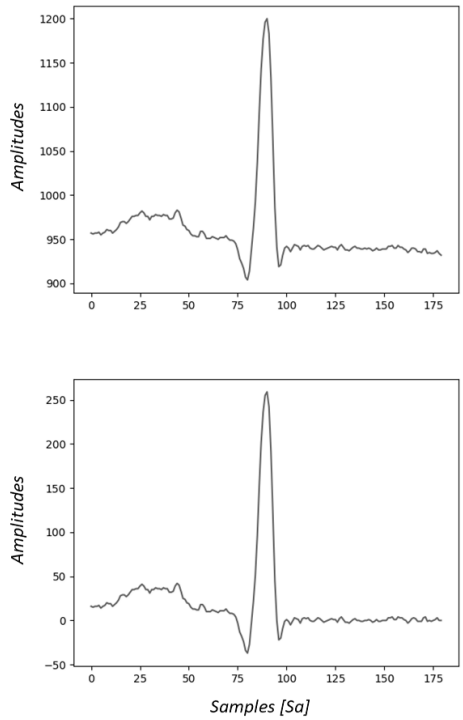

Figure 6.16: An example of a raw beat (on top) and the same beat with the 2-step median filter applied.

is computed as the contribution of a feature to maximize the split criterion used by the algorithm, also defined as the minimization of the impurity of child nodes, *i.e.,* Gini impurity [27].

In this way, starting from an initial set of 160 features, we selected only 39 and filtered the data set accordingly.

- RQ_2**: NEAPOLIS Accuracy**:
  With the purpose at answering RQ_2, we first evaluated the accuracy of *NEAPOLIS* by using different Machine Learning algorithms such as Random Forest [101], Support Vector Machine [174], k-nearest neighbors [51] and Multi-layer Perceptron [100]. In addition, we distinctly involved in

Table 6.11: Stratified split of the data set used for the classification experiment.

Beat type	**DS1**	**DS2**
APB	1,269	1,269
LBBB	4,023	4,023
N	37,109	37,109
RBBB	3,606	3,607
PVC	3,440	3,440
Total	49,447	49,448

the experimentation two consolidated state of the art approaches for handling with the problem of data unbalance. Specifically, we used (i) SMOTE [35], which makes an over-sampling of the minority class by creating synthetic minority class examples and (ii) Tomek's links, an undersampling techniques presented by Tomek [227]. We also tested standardization and scaling techniques based on the type of classifier used. For example, we used standardization with Support Vector Machine and *min-max* scaling with Random Forest.

Once identified the best configuration for *NEAPOLIS*, we compare its accuracy with our baseline [181]. The two approaches have been compared by using the Sensitivity, Specificity and Precision.

As for the validation, we followed the same protocol as the one proposed in our baseline [181], *i.e.,* a stratified split of the data set in two sub data sets, namely *DS1* and *DS2*. The result of the stratified split procedure is that both *DS1* and *DS2* contains a proportional number of instances, based on classes (*i.e.,* the beat types). Such a decomposition of the data set is depicted in Table 6.11.

In this way, we obtained two sub data sets where *DS1* was used for training and *DS2* for testing only. According to the validation protocol exhibited by Pandey *et al.* [181], the training set in turn was further split in 80% and 20% for a preliminary validation. In this way, for each model, in the training phase it is performed a preliminary validation on *DS1*. Then, the final testing was performed on *DS2*.

To avoid any convenient split of the original data set into *DS1* and *DS2*, we have repeated the splitting process several times, in order to have results less affected by the randomness. Especially, we selected 1,000 random seeds and then for each seed we repeated (i) the stratified split in *DS1* and *DS2* and (ii) the individual split of *DS1*. This means that we chose to repeat the complete validation protocol for 1,000 times and average the results accordingly.

Analysis of the Results

This section describes the results achieved aiming at answering our research questions.

RQ_1: Feature Analysis The main results of the experiment conducted to answer RQ_1 are depicted in Figure 6.17. We used a Random Forest classifier with a threshold of *1.25 * median* of the features importance. Specifically, in the figure we exhibit the five features with the highest weight.

In details, we obtained that the feature with the highest weight is the first reflection coefficient from the AR model. Almost with the same weights, we can find the fourth descriptor from the MODWT model and the *pre-RR interval*. Finally, the first and third coefficients, from the FFT, are also included in the top-5 ranking.

RQ_2: NEAPOLIS Accuracy As designed, we experimented several machine learning techniques to identify the best configuration for *NEAPOLIS*. The best configuration found is the one composed of *SMOTE*, *min-max scaler* and *Random Forest*, this latter set with 100 estimator trees. The classification accuracy achieved by *NEAPOLIS* using such a configuration is reported in Table 6.12. It is worth noting that such a configuration of *NEAPOLIS* is used for the comparison with our baseline.

Table 6.13 reports the comparison—in terms of overall accuracy—between *NEAPOLIS* and the selected baseline. Considering the average of the overall metrics, *NEAPOLIS* outperforms the state of the art baseline method in terms of sensitivity, specificity, precision and F1 score. In particular—with regards

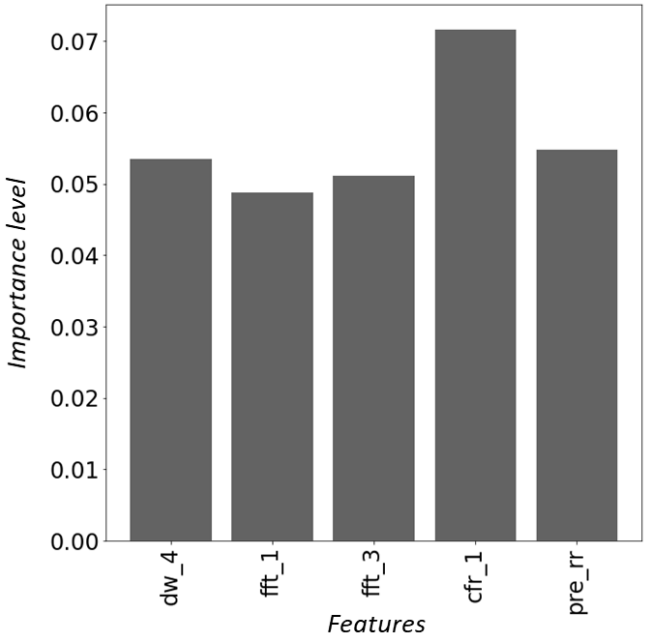

Figure 6.17: Top five selected features using importance weight.

to the sensitivity and F1 score—the improvement is greater than 2% and 1% respectively.

Performing a class level analysis (see Table 6.14), what emerges from the classification results is that *NEAPOLIS*, with regard to the *LBBB* class, shows an improvement greater than 1% and 0.5% only for Sensitivity and F1-score respectively while for the other metrics the results are almost the same. As far as *RBBB* class, *NEAPOLIS* shows a slight improvement for all the classification metrics except for the Precision that has a delta greater than 0.5%. *PVC* Class is the only one that has registered a decrease—that however does not exceed 0.05%—in terms of Specificity and Precision. On the contrary, *NEAPOLIS* shows a significant impact in terms of Sensitivity and F1 score for the same class, *i.e.,* greater than 3% and 1% respectively. With respect to the *APB* class, the improvement of *NEAPOLIS* is not significant in terms of Specificity and Precision

Table 6.12: *NEAPOLIS*'s classification metrics computed on the validation set *DS2*. Those values are averaged among the 1,000 runs of our validation protocol.

Beat type	**Sensitivity**	**Specificity**	**Precision**	**F1**
APB	90.48	99.81	92.49	91.47
LBBB	98.53	99.96	99.50	99.01
N	99.34	98.29	99.43	99.39
RBBB	99.18	99.97	99.68	99.43
PVC	98.28	99.61	95.02	96.62
avg	97.16	99.53	97.22	97.18

Table 6.13: Comparison of *NEAPOLIS* with the chosen baseline [181] in terms of Sensitivity, Specificity, Precision and F1 score.

Avg metrics	***NEAPOLIS***	**(Pandey et al., 2020)**	**Delta**
Sensitivity	97.16	94.89	+ **2.27**
Specificity	99.53	99.14	+ **0.39**
Precision	97.22	96.73	+ **0.49**
F1 score	97.18	95.77	+ **1.41**

but very high in terms of Sensitivity and F1 score, *i.e.,* equal to 7% and greater than 4%, respectively. Finally, for what concerns the N class, *i.e.,* the normal heart beats, *NEAPOLIS* outperforms—even slightly—the baseline method in terms of all the classification metrics.

Final Remarks

The main advantage of *NEAPOLIS*—with respect to state of the art tool—is that it can be easily involved in online scenarios of modern IoMT systems. Indeed, the proposed approach embeds only features that allow to obtain a prompt early diagnosis of arrhythmia conditions. In few words, *NEAPOLIS* does not embed features that need a long-term buffering and elaboration of the ECG.

As part of our future agenda, we aim at improving the validation technique by involving a scheme that avoids the random split, *i.e.,* that separates the data between train and test belonging to the same subject. In addition, we will try to improve the accuracy of *NEAPOLIS* by performing a fine tuning of the pa-

rameters of the machine learning models. We also plan to experiment Artificial Neural Networks as machine learning technique.

Table 6.14: Comparison of *NEAPOLIS* with the chosen baseline [181] at class level in terms of Sensitivity, Specificity, Precision and F1 score.

Class N			
Metrics	***NEAPOLIS***	**(Pandey et al., 2020)**	**Delta**
Sensitivity	99.34	99.31	+ **0.03**
Specificity	98.29	96.45	+ **1.84**
Precision	99.43	98.84	+ **0.59**
F1 score	99.39	99.07	+ **0.32**
Class LBBB			
Metrics	***NEAPOLIS***	**(Pandey et al., 2020)**	**Delta**
Sensitivity	98.53	97.52	+ **1.01**
Specificity	99.96	99.92	+ **0.04**
Precision	99.50	99.05	+ **0.45**
F1 score	99.01	98.28	+ **0.73**
Class RBBB			
Metrics	***NEAPOLIS***	**(Pandey et al., 2020)**	**Delta**
Sensitivity	99.18	98.97	+ **0.21**
Specificity	99.97	99.93	+ **0.04**
Precision	99.68	99.05	+ **0.63**
F1 score	99.43	99.01	+ **0.42**
Class PVC			
Metrics	***NEAPOLIS***	**(Pandey et al., 2020)**	**Delta**
Sensitivity	98.28	95.18	+ **3.10**
Specificity	99.61	99.63	-0.02
Precision	95.02	95.07	-0.05
F1 score	96.62	95.13	+ **1.49**
Class APB			
Metrics	***NEAPOLIS***	**(Pandey et al., 2020)**	**Delta**
Sensitivity	90.48	83.48	+ **7.00**
Specificity	99.81	99.79	+ **0.02**
Precision	92.49	91.64	+ **0.85**
F1 score	91.47	87.37	+ **4.10**

6.4 Analysis in Compressed ECG

Thanks to the Internet-of-Medical Things (IoMT), there has been a huge spread of wearable devices for healthcare applications. Machine Learning (ML) and Compressed Sensing (CS) are some of the tools involved in this kind of systems. Indeed, at the end of the first decade of 2000, Compressed Sensing has been investigated for healthcare applications [61, 17, 31].

6.4.1 Motivations

ECG signals need to be acquired and represented with high accuracy in order to ensure accurate signal analysis. Therefore, parameters such as the sampling rate and the accuracy of the electrocardiographic sample values must be able to allow for this requirement. This involves the generation of a signal with a considerable weight. Think for example of modern wearable devices or automatic analysis systems for decision support: a large signal impacts both the wireless transmission and the occupied server-side memory. Consequently, many efforts have been devoted by the scientific literature to define methods for ECG compression. In the context of this thesis work, to optimize the process of electrocardiographic data analysis, several approaches have been proposed that aim to derive diagnostic information from a compressed ECG signal.

The theory behind the Compressed Sensing basically states that when dealing with specific signals, satisfying the condition to be sparse in some domain, they can be reconstructed from a smaller number of samples, than that the actually needed by Nyquist rate sampling. The reconstruction phase from the compressed samples usually involves complex algorithms with a relatively high computational cost that should be able to guarantee the signal integrity by keeping clinically relevant features [54], [1]. This cost is balanced by the very small payload of the wirelessly transmitted data provided by the wearable device that performs data compression. In general, the CS methods require low computational effort for the compression, which is performed on the wearable device, and high computational effort for the reconstruction, which is usually performed on a more powerful device (e.g laptop), [140, 12]. In the recent years, several methods for the automatic detection of cardiac diseases from an ECG trace

and, more specifically, automatic classification of heartbeats has been proposed [159, 116, 193, 36, 58, 57, 77, 240, 153]. All these methods — when applied in contexts of long-term continuous monitoring — require that the physical devices of the IoMT system continuously send the monitored data to a gateway or directly to a server. Since wearable devices are battery-powered and since they use a wireless connection to exchange data, it is very important to decrease the amount of transmitted data for reducing the energy consumption of the device and, as a consequence, increasing its battery life. [13] proposed an ECG data acquisition system that performs data compression according to compressed sensing, a theoretical framework that exploits the sparsity of a signal in a specific domain without computationally load the physical device that performs the compression. The authors also showed that — by using such kind of systems — it is possible to reduce the number of transmitted data and increase the battery life by $\sim$12, while keeping a good ECG signal reconstruction quality.

6.4.2 Identification of R-Peak occurrences

Not all the ECG features are relevant for some clinical evaluation, thus in such applications it is not needed to reconstruct the entire ECG signal [139, 209]. For example, in case of the most well-known cardiac arrhythmia, the Atrial Fibrillation, there is a huge literature dedicated to the process of automatic detection. For this purpose, there are works which take into consideration both the morphological and rhythmic aspects of the ECG, such as the one proposed in [139, 209] but there are also many works which implement a very accurate detector of Atrial Fibrillation [249, 185], where the base feature is only the Heart Rate.

The detector proposed in this section aims at identifying the number of R-peaks, by means of Machine Learning, in a signal obtained from a compressed version of a windowed portion of a raw single-lead ECG. Our study has been conducted on two types of signal, both outputs of the CS method described by Balestrieri *et al.* [14]: (i) the first one obtained from a 1-bit quantization process, and (ii) the second one obtained from the multiplication of a sensing matrix with the original signal.

The proposed detector presents the advantage of not requiring the reconstruction of the signal. In addition, this approach aims at identifying the R-peaks

occurrences with a high Compression Ratio, by keeping comparable accuracy in the classification process, with respect to the state of the art. Furthermore, in case the reconstruction of the entire ECG would be needed, the domain of the CS method in [14] allows to obtain a better reconstructed signal. Finally, the detector involves the use of Machine Learning predictive models. This represents another advantage because such models may have a very small computational cost. In this section, the key detection is focused on the identification of R-peaks. The R-peak occurrences can represent a crucial information of an ECG, which can lead to a highly precise estimation of the heart-rate.

As baseline method, we chose the work presented by Da Poian *et al.* [54], that represents one of the first and best approaches of Information Retrieval in compressed signals. The authors considered a framework where the ECG signals are represented under the form of CS linear measurements. The QRS locations have been estimated from the compressed signal by computing the correlation of the compressed ECG and a known QRS template. The results show that this solution is competitive with methods applied to the reconstructed signal. Therefore, the complex and time-consuming steps of pre-diagnosis and diagnosis — usually manually performed by specialized medical staff — can be supported or undertaken by such systems.

Our R-peak occurrences detector proposed in this section has been tested on two types of compressed data provided by the CS dynamic sensing scheme proposed in [14]. This CS method is based on a sensing matrix, which depends on the power of each frame of the ECG signal. In particular, it provides a vector $\mathbf{y}$ of M samples that is a compressed version of the vector $\mathbf{x}$ of N ECG samples. In the following, a brief description of the CS-based method [14] is reported. Let us consider the vector $\mathbf{x}$, a frame of N samples of the ECG signal. The value x_{avg} is obtained as the average of $\mathbf{x}$. Then, the vector $\mathbf{x}_a$ is evaluated as follows:

$$\mathbf{x}_a = \mid \mathbf{x} - x_{avg} \mid \tag{6.2}$$

According to a threshold x_{th}, which is chosen experimentally, the Power Information Vector (PIV) $\mathbf{p}$ is constructed. This vector contains the values of one at the indices where the vector $\mathbf{x}_a$ exhibits values higher than x_{th}, and zeros, otherwise.

The sensing matrix $\mathbf{\Phi}$ is defined as a circulant matrix, where the first row is the vector $\mathbf{p}$:

$$\mathbf{\Phi} = \begin{bmatrix} p(1) & p(2) & \dots & p(N) \\ p(N-CR+1) & p(N-CR+2) & \dots & p(N-CR) \\ \vdots & \vdots & \ddots & \vdots \\ p(CR+1) & p(CR+2) & \dots & p(CR) \end{bmatrix} \tag{6.3}$$

where, $CR = \frac{N}{M}$ is the compression ratio. The vector $\mathbf{y}$, containing the compressed data, is given by the multiplication between the sensing matrix $\mathbf{\Phi}$ and $\mathbf{x}$. All those processing steps are intended to be performed by a wearable device, battery powered and wirelessly connected to Internet. Thus, it sends the vectors $\mathbf{p}$ and $\mathbf{y}$ to another device, having much more processing capabilities (e.g. a laptop), for performing the signal reconstruction of $\mathbf{x}$. In particular, $\mathbf{p}$ contains information related to the power of the acquired N samples and has a size of $N/8$ bytes, and $\mathbf{y}$ contains the M compressed samples and has a size of $b \cdot M/8$ bytes, where b is the number of bits used for representing an ECG sample. From those two vectors, the reconstruction of the signal $\mathbf{x}$ is performed as follows. At the beginning, from the vector $\mathbf{p}$, the sensing matrix $\mathbf{\Phi}$ is constructed (6.3). The dictionary matrix $\mathbf{\Psi}$ is defined according to the Mexican hat wavelet kernel, which has been considered to define the domain where the signal $\mathbf{x}$ is sparse;

$$\mathbf{\Psi} = \left[\mathbf{\Psi_{base}}, u\right] \tag{6.4}$$

where, u is a vector of N ones, and $\mathbf{\Psi_{base}}$ is given by:

$$\begin{aligned} \mathbf{\Psi_{base}} = \Bigg[& \psi(2,0), \psi(2,2), \psi(2,4), \dots, \psi\left(2, 2 * \frac{N-1}{2}\right), \\ & \psi(4,0), \psi(4,4), \psi(4,8), \dots, \psi\left(4, 4 * \frac{N-1}{4}\right), \\ & \dots, \psi(N,0) \Bigg] \end{aligned} \tag{6.5}$$

with:

$$\psi(a,b) = \frac{2}{\sqrt{3a}\cdot\pi^{1/4}}\cdot\left[1-\left(\frac{\mathbf{n}-b}{a}\right)^2\right]\cdot e^{-\frac{1}{2}\left(\frac{\mathbf{n}-b}{a}\right)^2} \tag{6.6}$$

with $\mathbf{n} = [0,\ldots,N-1]^T$. By knowing the matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$, the Orthogonal Matching Pursuit (OMP) algorithm is performed to estimate the α coefficients vector, which represent the sparse coefficient of the signal in the Mexican hat domain. Finally, once the α is available, the reconstructed signal $\hat{\mathbf{x}}$ is obtained as follows:

$$\hat{\mathbf{x}} = \mathbf{\Psi}\cdot\alpha \tag{6.7}$$

In order to detect the R-peaks occurrences, it would be needed to perform the reconstruction and then to apply any kind of detector. The reconstruction is performed by OMP, which exhibits a computational complexity of $O((N+M)S)$, where, S is the number of iterations of the OMP algorithm, which is in any case lower than N. The idea underlying this work is not performing the OMP reconstruction, but executing the R-peak detector algorithm directly on the compressed vector $\mathbf{y}$ or the PIV, $\mathbf{p}$.

For identifying R-peak occurrences, the proposed ML-based method has been implemented by considering separately as input the two signals provided by the CS method previously described: in the ***PIV*** version of the proposed tool, the input of the classifier is the signal resulting from the process of 1-bit quantization, the PIV, $\mathbf{p}$; while, in the ***CS*** version, the input is the signal resulting from the CS process, $\mathbf{y}$.

The *PIV* and the *CS* detectors basically differ in the technique of Digital Signal Processing expected in the workflow. This is depicted in Figure 6.18. The final stage of classification is handled by a Machine Learning classifier which provides the number of R-peaks in the windowed segment of the signal.

The Machine Learning classifier chosen in this preliminary phase of the study is the *Random Forest*, first proposed by Breiman [26]. A random forest basically represents a combination of tree predictors. In this context, a tree is defined as a function of a randomly initialized vector. This vector follows these properties: (i) it is sampled independently and (ii) with the same distribution as all the other

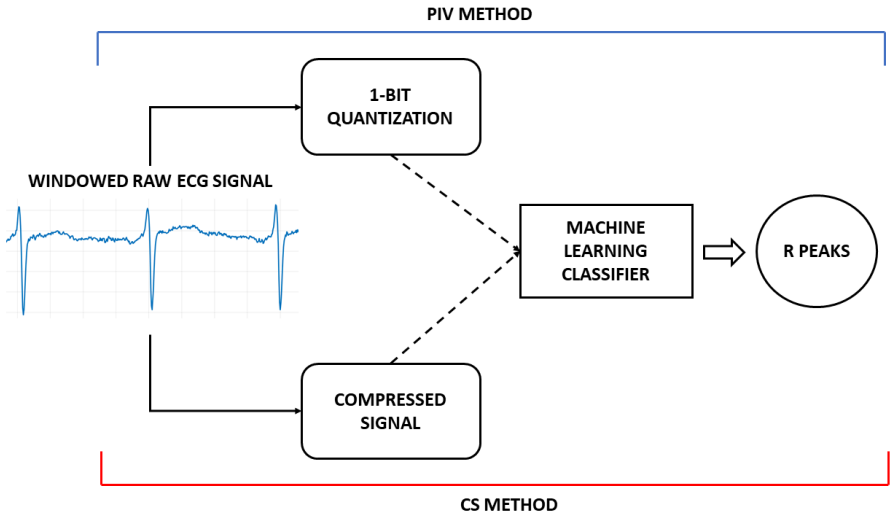

Figure 6.18: The different workflows for the two proposed versions of the R-peak occurrences detector.

trees in the forest. In other words, a random forest integrates trees and each of them provides a class prediction, as outcome.

The class with the highest number of votes represents the prediction of the entire model. Breiman in [26] defines the algorithm as follows:
"a random forest is a classifier consisting of a collection of tree-structured classifiers $h(x, \Theta_j), j = 1, ...$ *where the* $\mathbf{\Theta_j}$ *are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input* $\mathbf{x}$*"*.
Thus a Random Forest represents a tree-based ensemble with each tree depending on a collection of random variables. More formally, suppose given a k-dimensional random vector:

$$\mathbf{A} = (\mathbf{A_1}, ..., \mathbf{A_k})^{\mathbf{T}} \tag{6.8}$$

representing the input, and a random variable $\mathbf{Y}$, representing the output. In this case, assuming an unknown joint distribution $\mathbf{T_{AB}}(\mathbf{A}, \mathbf{B})$, the goal – as expected by the algorithm – is to find a prediction function $\mathbf{f}(\mathbf{A})$ for predicting

$\mathbf{B}$. The function $\mathbf{f}(\mathbf{A})$ is determined by a loss function $\mathbf{L}(\mathbf{B}, \mathbf{f}(\mathbf{A}))$ and defined to minimize the expected value of the loss:

$$\mathbf{E_{AB}}(\mathbf{L}(\mathbf{B}, \mathbf{f}(\mathbf{A}))) \tag{6.9}$$

where the subscripts denote expectation with respect to the joint distribution of A and B. In the classification scenario, if the set of possible values of $\mathbf{B}$ is denoted by $\mathbf{B'}$, minimizing $\mathbf{E_{AB}}(\mathbf{L}(\mathbf{B}, \mathbf{f}(\mathbf{A})))$ for zero-one loss gives:

$$\mathbf{f}(\mathbf{a}) = \underset{\mathbf{bB'}}{\operatorname{argmax}}[\mathbf{T}(\mathbf{B} = \mathbf{b}|\mathbf{A} = \mathbf{a})] \tag{6.10}$$

otherwise known as the Bayes rule [52].

Thus intuitively, the Random Forest algorithm expects to divide the source data in a random number of subsamples. For each of these – based on a random set of features – a decision tree is built. The final prediction is taken depending on the individual votes, which fall into leaves. The results, from each individual tree, are gathered and avereged. An example of such a procedure, is depicted in Figure 6.19. In the classical Breiman implementation, the training dataset covers around the 63 % of the total data while the remaining (approximately the 37 %) is used to validate the ensemble of the decision trees, in other words the model.

Empirical Evaluation

In this Section, details about the experiments conducted to compare and validate the two versions of the proposed tool, are given. The dataset used for the assessment is the Physionet [82] MIT-BIH Normal Sinus Rhythm Database[5]. This database includes 18 long-term ECG recordings of subjects referred to the Arrhythmia Laboratory at Boston's Beth Israel Hospital. The ECG recordings belonging to this database do not present significant arrhythmia episodes; they include 5 men (aged 26 to 45) and 13 women (aged 20 to 50).

To compare the results of the presented approach (working on the compressed data), the Pan-Tompkins method [178] (working on the original signal) has been used as state of the art. Indeed, it is largely considered as the QRS detector of

[5]https://physionet.org/content/nsrdb/1.0.0/

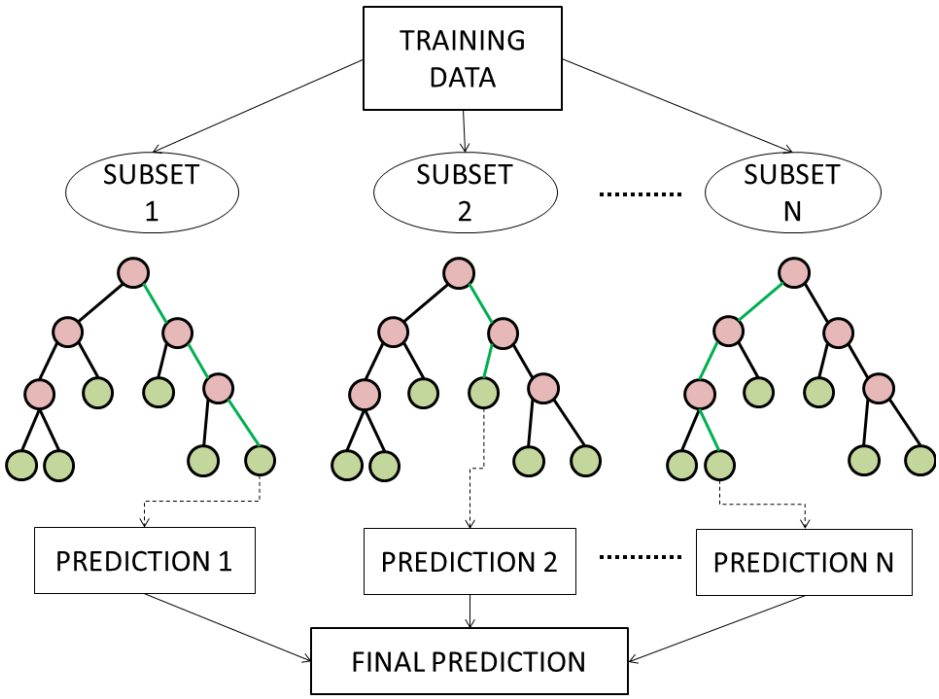

Figure 6.19: Example of a Random Forest workflow.

reference in the literature. The setup phase for the training and testing of the model has consisted of the following choices:

- *Feature Alignment*: an observation window with a fixed length of 2, has been chosen. Thus, considering a sampling frequency of 128, a portion of 256 samples is obtained from a raw ECG signal.
- *Class labels*: for a supervised experiment, a 3-class classification model has been considered. More specifically, each class contains the information related to the R-peak occurrences in each of the 2 ECG segments. The 3 classes chosen in this part of the study are: (1) for 1 single R peak, (2) for 2 R-peaks and (3) for 3 R-peaks.
- *Validation of the model*: to evaluate the accuracy of the presented approach, a classical Leave-1-Person Out (L1PO) cross-validation has been used. The

data has been decomposed in n folds, one for each subject. Then, each of such folds has been used as test set and the union of the remaining folds as training set. This type of validation process has the effect that the data related to a single patient were embedded once in the test dataset and *n-1* times in the training dataset. This technique allows to build a classifier which is not trained and tested on the data belonging to the same patient. It has been chosen to evaluate the model in the most challenging scenario: a patient-independent detection tool.

- *Number of instances*: due to the long-lasting records in the NSR database and the costly validation, the proposed choice was to work with a sample size having 95 of confidence level and 5 of confidence interval, with respect to the population. In this case, the population is a long-term ECG record. Considering that each record lasts 24 hours, there is a population of around 43200 2 segments. Thus, for the specific purpose of this study, around 380 instances have been randomly selected for each of the records belonging to the dataset. The selection of the instances respected the representativeness of the class label for each population.

Analysis of the Results

To evaluate the classifier, the metrics 11-14 have been considered for class-specific analysis (where each of the possible classifications is evaluated, e.g. class for occurrence equal to 1, 2 and 3), while metrics 15-16 for global analysis:

$$PRECISION = \frac{TP}{TP + FP} \tag{6.11}$$

$$RECALL = \frac{TP}{TP + FN} \tag{6.12}$$

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{((S_P) * (TP + FN) * (TN + FP) * (S_N)}} \tag{6.13}$$

$$F1_{score} = \frac{2 * TP}{(2 * TP) + FP + FN} \tag{6.14}$$

Table 6.15: Detailed global metrics evaluated for each class.

Metrics by class	Precision	Recall	MCC	F_1score
R-peaks = 1	0.361	0.898	0.564	0.515
R-peaks = 2	0.919	0.906	0.826	0.912
R-peaks = 3	0.913	0.904	0.818	0.908

Table 6.16: Performance comparison in terms of global metrics of the method based on PIV and Pan-Tompkins.

Method	Accuracy	ICI
PIV	0.905	0.095
Pan-Tompkins	0.928	0.072

$$ACCURACY = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.15}$$

$$ICI = \frac{FP + FN}{TP + TN + FP + FN} \tag{6.16}$$

where TP, FP, TN, FN indicates "correctly classified", "incorrectly classified", "correctly rejected" and "incorrectly rejected", respectively; *ICI* is the "Incorrectly Classified Instances" parameter, S_P is the sum of positives and S_N is the sum of negatives.

Proposed method with PIV For the evaluation of this detector, a fixed threshold of 0.3 in the process of 1-bit quantization has been used, see Figure 6.20. This is the x_{th} defined and presented between equations 1 and 2. In Table 6.15 the class-specific metrics are reported. The main outcome from Table 6.15 is that the classification is more accurate when dealing with ECG windows containing 2 or 3 R-peaks. Class 1 presents a high loss in terms of precision. As indicated, the Pan-Tompkins method, applied to the original raw signal, was the reference to the state of the art. For such a purpose, the work presented in [205] has been used. The comparison between the performances obtained by the *PIV* method and the Pan-Tompkins approach are reported in Table 6.16.

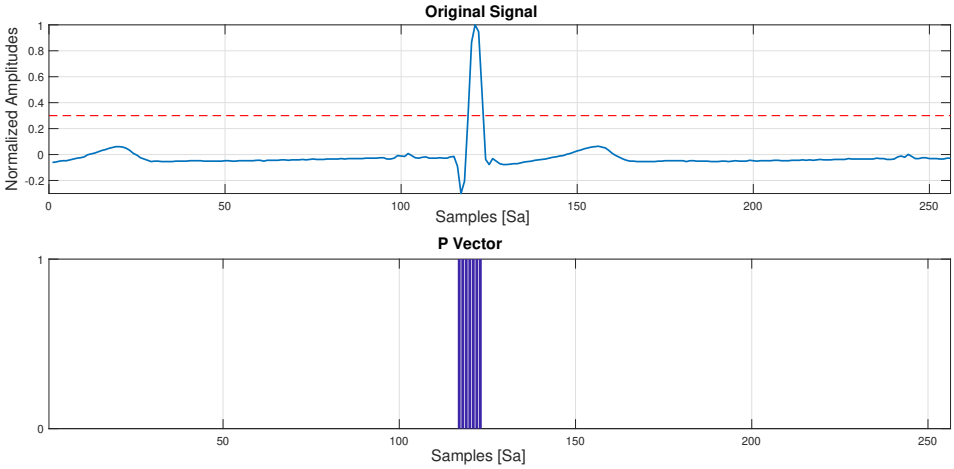

Figure 6.20: Example of a signal submitted to the process of 1-bit quantization.

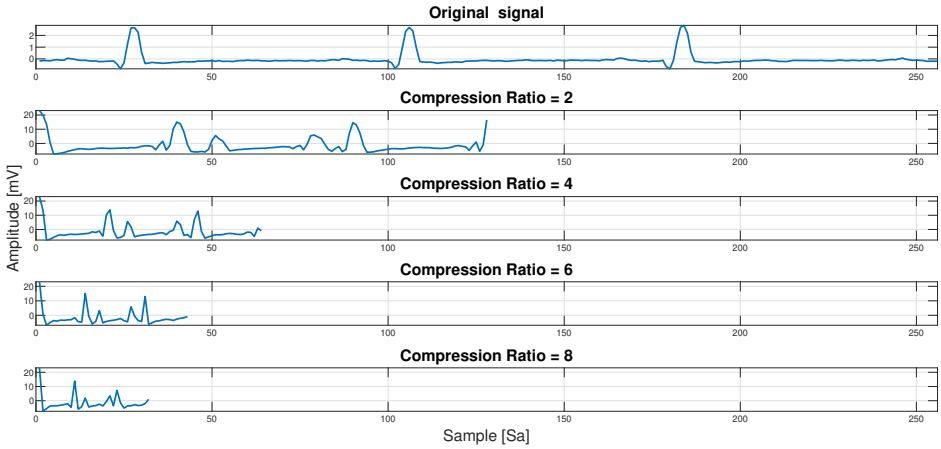

Figure 6.21: An example of the compressed samples contained in $\mathbf{y}$ for several values of CR.

The outcome of this experiment mainly reveals that the detection in the *PIV* stage is highly comparable with a milestone in the state of the art.

Table 6.17: Performances comparison in terms of global metrics at different CR.

CS Method	Accuracy	ICI
$CR = 2$	0.743	0.257
$CR = 4$	0.741	0.259
$CR = 6$	0.727	0.273
$CR = 8$	0.726	0.274

Proposed method with compressed samples For this version of the proposed approach, the performance at 4 different CR have been investigated. The highest ratio evaluated is 8, which means a 32 samples input to the classifier. An example of compressed signals at several CR is shown in Figure 6.21.

The results obtained in this study are reported in Table 6.17.

The loss in terms of global accuracy with respect to the state of the art becomes significant in this study. What emerges here is that the accuracy in the identification of R-peaks does not highly depend on the compression ratio, considering the subset $2, 4, 6, 8$. This experiment has shown that, when dealing with applications of R-peak occurrences detection, the Power Information Vector embeds much more information with respect to the compressed signal. Thus, for this specific type of detection, the quantization stage provides better results than the Compressed Sensing full algorithm. However, the compressed signal does not have any knowledge of the $\mathbf{\Phi}$ matrix. In other words, this result can be due to the fact that the compressed signal does not embed the information on how it has been obtained from the original signal.

Final Remarks

The results show that the *PIV* detector is highly comparable with the most used method in the state of the art, the Pan-Tompkins approach, with a loss of approximately 2 on the global accuracy. The *PIV* detector shows comparable results even when compared with a similar approach in the state of the art, the one by Da Poian *et al.* [54]. In this work, the authors report a sensitivity of around 91 % when dealing with a signal compressed with a CR approximately equal to 6.67. The *PIV* detector also shows similar values (approximately 90 % but with the possibility to reach values of CR equal to 12. On the other hand, by

using the detector with the compressed signal as input, the global loss – in terms of accuracy – assumes a significant amount. The huge advantage in building a detector, such as the one proposed in this work, relies on (i) the very low-cost elaboration to obtain the *Power Information Vector* in the CS domain and (ii) on the benefit that – even if there is a specific need to work with the full ECG signal – with a single *PIV* it is possible to reconstruct up to 12 ECG signals. As future works, many studies can be further conducted, such as (i) trying to use different categories of classifiers – specifically, the LSTM Recurrent Neural Network, highly suited in case of time-series – and (ii) trying to detect cardiac pathologies directly in the CS domain.

6.4.3 Heartbeat Classification

In this section, we describe *RENEE* (hea**R**tb**E**at classificatio**N** in compr**E**ssed **E**CG), a novel method for the automatic classification of heartbeats that works on a compressed ECG signal, through the involvement of a method based on a 1-bit signal quantization [186]. The advantage of using *RENEE* with respect to others available heartbeat classifiers is that it allows reducing the signal data rate and performs the heartbeat classification directly on the quantized samples, therefore without reconstructing the signal waveform. Especially, *RENEE* was designed to be used in an IoMT system based on wearable devices, where it is needed to reduce the data transmitted by the physical device and to automatically classify the heartbeats, without analyzing the signal waveform. In such a context, using a compressed signal is beneficial both in terms of data rate and memory occupied.

The pre-processing stage of *RENEE* is composed by an R-peak detection algorithm and a consecutive selection of a complete heartbeat signal. Once executed these steps, the further processing consists in the compression. Finally, the compressed data is provided as input to the machine learning classifier. This latter is in charge of providing the final *multi-class* classification on the heartbeat types. We provide more details about each step of our approach below.

Pre-processing of ECG Data The pre-processing steps expected from our proposed approach are composed of an *R-Peak detector* and a *heartbeat selection technique*. Such steps may be performed on the transmitter device.

The *R-peak detector* is in charge of accurately evaluating the R-peak positioning in a single-lead ECG signal. For our purposes, we used the R-peak annotations available from the database but — for an online scenario — a R-peak detection algorithm has to be involved in *RENEE*, such as the Pan-Tompkins algorithm [180, 206].

The *heartbeat selection technique* needs the R-peak positioning information provided by the previous algorithm in order to properly select the heartbeat. According to the chosen baseline, a heartbeat signal is defined as the samples included between two middle points of three successive R-peaks. In other words, a heartbeat is not computed as an ECG signal included between two R-peaks, but as a signal composed of (i) an individual QRS complex, and (ii) the previous and successive dynamics.

After all these steps, we compressed the data through the method proposed by PICARIELLOmeas, based on a 1-bit signal quantization.

Features Creation & Classification In order to create informative features for the classification stage, we first defined a windowed accumulation of samples by imposing a fixed window length *winLen*: given the *qhbs* signal, we define a new signal *whbs* so that $whbs_i = \sum_{j=i-winLen}^{i} qhbs_j$. In this way, *RENEE* is able to represent the dynamics of the original heartbeat signal in the compressed domain.

The classification component of *RENEE* — using of machine learning techniques — is in charge of providing the final classification of the heartbeat in five types, according to the AAMI standard ec571998testing: normal beat (N), ventricular ectopic beat (V), supraventricular ectopic beat (S), fusion of a normal and a ventricular ectopic beat (F) and unknown beat type (Q).

The features we use for the automatic classification of the heartbeats are the samples of the final signal we obtained, *i.e., whbs*. It is worth noting that the number of samples may vary among different heartbeats. However, the machine learning model, appointed for the classification of each heartbeat signal, needs the data to be aligned in terms of features across all the instances. Therefore, we needed to select a maximum number D of *whbs* samples to use as features. To do this, we used the same method used by Xu *et al.* [240], *i.e.,* we apply

zero-padding and truncation in case the heartbeat signal contains less or more samples as compared to a fixed threshold D, respectively.

Figure 6.22 depicts an example of how the signal changes after each of the main stages of *RENEE*. The plot in the upper row shows the original signal waveform of a sample heartbeat. The second subplot shows the signal after the application of dithering noise. Then, the samples obtained by the 1-bit quantization procedure are depicted in the third subplot. Finally, the fourth line of plot shows the signal once applied the windowed accumulation of samples. Such a signal contains the features used by the machine learning method for the classification of the heartbeat.

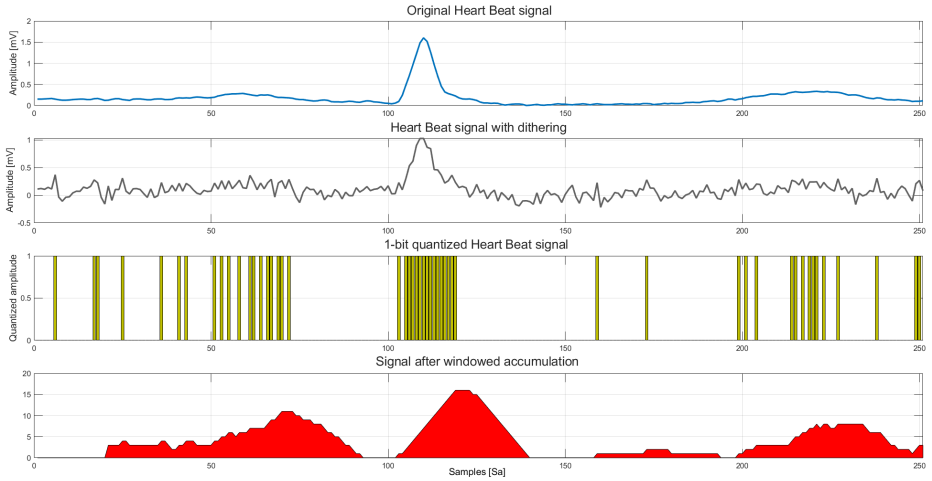

Figure 6.22: The four main steps performed by *RENEE* after the pre-processing stage: (1) the original heartbeat signal, (2) the signal after the pre-processing and noising, (3) the compressed signal through 1-bit quantization and (4) the final elaboration — applied to the compressed signal — that consists of a windowed accumulation of binary samples.

Empirical Evaluation

The *goal* of this study is to evaluate the accuracy of *RENEE* in classifying heartbeat types from a highly compressed version of an ECG. The literature

shows that the uncompressed trace of an ECG allows to obtain a very accurate classification [240]. Thus, the study is steered by the following research question:

Can RENEE provide a heartbeat classification comparable to state-of-the-art methods based on uncompressed ECG?

The *perspective* of the study is both (i) of a researcher who wants to understand if machine learning techniques can classify heartbeat also in the compressed domain, and (ii) of a practitioner who wants to use a method in a telemedicine application that is able to balance accuracy and data storage and transmission.

Context of the Study The context of this study is represented by the MIT-BIH Arrhythmia Database [160, 84].

Each heartbeat is classified by using 15 different classes. These 15 types of heartbeat in the MIT-BIH arrhythmia database have been categorized in five classes, reported in [5].

Table 6.18: Clustering of the original heartbeat types in five groups according to the ANSI/AAMI EC57:1998 [5].

AAMI Class				
N	**S**	**V**	**F**	**Q**
NOR	AP	PVC	fVN	P
LBBB	aAP	VE		fPN
RBBB	NP			U
AE	SP			
NE				

In most of the recordings in the MIT-BIH database, the first channel is a modified limb lead II (MLII), and the second one is a modified lead V1. In our experiments, only the signal from the first channel was used for ECG classification because, typically, QRS complexes are usually prominent [240].

Finally, according to the AAMI recommendation [5], we removed from the dataset four recordings containing paced beats. The final dataset was composed of a total of 44 records.

We experimented a large set of machine learning techniques to train the model embedded in *RENEE*. To execute a complete experimentation, we chose at least

one classifier from each category of classifiers available from the Weka machine learning toolkit [93], *i.e.,* J48 [190], Replication Tree [60], Random Forest [16], Logistic regression [50], AdaBoost M1 [69], BayesNet [6], J48 [44] and a 3-layer long short-term memory (LSTM) Neural Network (NN) [136]. We chose the Long Short-Term Memory (LSTM) NN (instead of DNN, for example) because ECG signals are time series data and LSTM is capable of learning long-term dependencies [240]. We have also included in our study several classifiers implemented in the Matlab Classification Learner app[7]. Basically, they are specific implementation of different categories of classifiers. Examples are the k-nearest neighbors (KNN)[56] and the Support Vector Machine (SVM)[174].

The whole dataset, composed by 44 records, was split into two datasets, *i.e.,* DS1 and DS2: this allows to perform a patient-independent classification in which each patient appears either in the training or in the test set, but never in both of them. Each dataset contains approximately 50,000 beats from 22 recordings. We used DS1 as the training set and DS2 as the test set. This is a consolidated procedure from the literature. Indeed, it was used in many previous works [59, 243, 192, 240].

As for the parameters of *RENEE*, we used the configurations reported in Table 6.19 and determined with a trial & error approach—on a different data set—than the one used for the classification experiment.

We compare *RENEE* with the chosen baseline work, *i.e.,* a state-of-the-art method designed to automatically classify uncompressed ECG heartbeats with high accuracy [240].

To compare the approaches, we use several metrics. First, we use the accuracy, *i.e.,* the number of correctly classified instances divided by the total number of instances. Then, we use also some class-level metric—designed for a given class—among the ones we consider for our study *i.e.,* N, S, V, F.

We report the comparison for the classes N, S, V, F. Similarly to other studies [59, 243, 192, 240], we exclude the class Q, because such a class contains paced beats (that were excluded) and unclassifiable beats (only 15).

[6] https://bit.ly/3bYCFcR
[7] https://bit.ly/35oMcZ9

Table 6.19: Configuration of *RENEE*'s parameters used in the experimentation.

Parameter	Description	Value
σ	Power of Gaussian dithering	0.1
γ	Threshold for the quantization	0.2
winLen	Window size for *whbs*	20
D	Number of features	417

Threats to Validity A limitation of this study may be represented by the validation. Even if our kind of validation takes care to appropriately separate the data of distinct subjects for the training and testing phases, a more appropriate validation would have been a typical L1SO-CV (Leave 1 Subject Out Cross Validation). This implicates that the data related to an individual patient will be included once in the test data set and *n-1* times in the training data set. However, we decided to adopt the validation method used in previous study to facilitate the comparison of the results achieved. The replication of the study on larger data sets and with different validation methods is part of the agenda of our future works.

Analysis of the Results

We report in Table 6.20 the accuracy of *RENEE* obtained by using the 10 top performing classifiers experimented in our study. The achieved results show that *RENEE* allows to keep a comparable overall accuracy in the classification of heartbeats when compared to the baseline, which, as previously mentioned, uses the uncompressed original ECG signal. Especially, the Random Forest, the Bagged Trees and the Medium Gaussian classifiers achieve the highest accuracy (between 0.93 and 0.94).

Table 6.21 reports the detailed results achieved by *RENEE* when using the best performing classifier, *i.e.,* Random Forest. The results reported in Table 6.21 highlight a clear outcome: *RENEE* can correctly classify with a high accuracy the classes labeled as *N* and *V*. In details, *RENEE* has correctly classified respectively 43,978 out of 44,259 and 2,713 out of 3,221 total instances. For what concerns the class *S*, *RENEE* still needs to improve in terms of classification accuracy. The

Table 6.20: Overall accuracy of *RENEE* by using the 10 top performing classifiers experimented in our study. At the bottom we also report the accuracy achieved by the approach proposed by Xu *et al.* [240].

Classifier	**Accuracy**
Random Forest	**0.940**
Bagged Trees	0.937
Medium Gaussian SVM	0.932
Boosted Trees	0.928
LSTM NN	0.928
Fine Gaussian SVM	0.926
Fine Tree	0.925
Quadratic SVM	0.922
JRip	0.919
Cubic SVM	0.912
Baseline [240]	0.947

machine learning model with the highest classification accuracy specific for class S is the Quadratic SVM; such model was able to correctly classifies 128 instances out of approximately 1,800 total instances. Finally, for the F class, the results obtained in terms of Precision, Recall and F-Measure are not satisfying. It is worth noting, however, that the classification performance on these classes has a low impact on the overall performance. Indeed, as suggested by the standard ANSI/AAMI EC57 [5], it is recommended to focus the attention on the two majority arrhythmia classes, *i.e.,* classes S and V.

Table 6.21: Detailed classification evaluation of *RENEE* when using the best performing classifier, *i.e.,* Random Forest.

Class	**Precision**	**Recall**	**F-Measure**	**AUC**
N	0.945	0.994	0.969	0.966
S	0.375	0.016	0.031	0.565
V	0.907	0.842	0.873	0.987
F	0.074	0.018	0.029	0.812

Table 6.22 shows the MCC achieved by the two compared approaches for each class. The achieved results indicate that *RENEE* achieves similar results (even

if it performs slightly better) compared to the baseline for the classes *N* and *V*. Indeed, for these classes, the difference in terms of MCC is below 0.05. The greatest delta has been obtained for the class *S*, which touches the amount of 0.6. Thus, further improvements are required for the classification in the compressed domain of this particular class.

Table 6.22: Comparison between *RENEE* (Random Forest) and the approach proposed by Xu *et al.* [240].

Class	**Xu *et al.* [240]**	***RENEE***	**Delta**
N	0.69	0.67	-0.02
S	0.67	0.07	-0.60
V	0.91	0.87	-0.04
F	0.22	0.03	-0.19

Final Remarks

An empirical evaluation conducted on the MIT-BIH Arrhythmia Database indicates that the overall classification accuracy of *RENEE* is comparable to the accuracy of the best state-of-the-art method for the classification of heartbeats based on uncompressed ECG signal (0.940 *vs.* 0.947). Future work will be devoted to replicate the evaluation of *RENEE* by using more robust validation, such as the L1SO-CV to corroborate our findings.

Part III

Biomedical Monitoring of software developers.

CHAPTER 7

Definition of a Framework for the Biomedical Monitoring

Contents

7.1 Introduction

The creator of the C++ programming language, Bjarne Stroustrup once said that "our civilization runs on software"[1]. This statement is strikingly supported by reality, in which software controls a huge variety of devices. In fact, it is hard to imagine an area of human activity that does not rely at least partially on software, including professional work, home activities, and entertainment.

[1] https://bit.ly/3kkIzc8

Software development can be roughly defined as a set of activities that includes tasks such as implementing new features, analyzing requirements, and fixing bugs[15] and it is therefore nowadays crucial for a large variety of aspects.

Quality assurance is an expensive yet important part of software development because—needless to say—software is not perfect. Thus considering the importance of software development, the prediction of defects in the software has been largely addressed by the research community[99, 112, 113].

This was done by analyzing code-based metrics and also environmental and attitudinal factors, such as the stress.

In large industrial projects, software developers have to carry out the tasks assigned to them by their project team leader. The coding tasks can be mainly categorized in *Implementation* and *Bug Fixing*, with the first one intended as the activity aimed at generate a new software feature and the second one aimed at finding and fixing the bug in a snippet or program.

In this chapter, we present an approach for defect prediction completely based on the monitoring of developers and therefore on what happens *while they complete a task*.

We first identified the developers' aspects that we need to monitor, *i.e.,* the ones that, in principle, could help determining the outcome of a development task: we defined a set of *behavioral*, *psycho-physical*, and *contextual* factors. Based on them, we defined a set of *candidate predictors*, *i.e.,* features that could be concretely measured given specific signals (such as the *heart-rate* or the *attention level*).

The complete workflow of our approach is depicted in Figure 7.1. As a first step, we measure the signals and the metrics we need to compute the features. Then, we extract the feature as previously described. Given a set of labeled data for which we have both (i) the features we use, and (ii) the task result (*correct* or *buggy*) we selected the best features and train a classifier and then we use the related model to predict the outcome.

To test the effectiveness of our approach, we conducted a controlled experiment involving 20 software developers. Our experimental protocol provided for the monitoring of developers while they performed a total of four coding tasks each, two *implementation* tasks and two *bug-fixing* tasks, of different difficulty

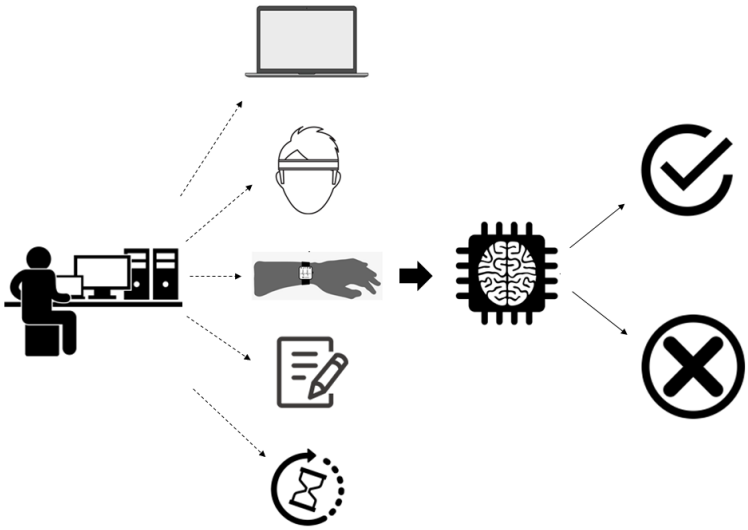

Figure 7.1: The workflows for the proposed approach.

levels. We derived the tasks from LeetCode[2]: this allowed us to automatically evaluate the correctness of each task by using large test suites provided by such a platform. We used the labeled data we collected to check the performance of our developer-based features. To this aim, we trained and tested a Multilayer Perceptron classifier in two different scenarios, and we compared our developer-based model with a classical code-based model, which focused only on code-related features.

7.2 Developers' Performance Factors

In this section we describe the approach we use to predict mistakes made by developers. First, we introduce the factors that we want to monitor during the task and we explain why such factors are relevant, in theory, for predicting the introduction of bugs while completing a task; then, we introduce the candidate

[2]A website where whoever can practice coding skills. There are many (and growing) questions, each with multiple solutions. Questions are ranked by difficulty level: easy, medium, and hard. (`https://leetcode.com`)

predictors we compute based on the signals we chose to monitor to capture the aspects in which we are interested;

7.2.1 Monitoring Factors

We focus on three families of factors: *activity-related*, (what they do during the task), *psycho-physical* (how they feel during the task), and *contextual* (what is the context in which they perform the task).

Behavioral Factors

What developers *do* during a task can provide some important hints that could allow to predict their success/failure in completing a specific task. Developers perform several activities while they write code: they type, they browse the web, they possibly get distracted by notifications. To capture such pieces of information, we focus on the developers' *activity* and on their *actions*.

We capture the high-level **activity** that the developer is performing (browsing the web, writing code, or something else). For example, if a developer uses the browser more than the IDE, it might be the case that he/she is experiencing some problem in understanding the problem at hand or how to use a specific technology.

The low-level **actions** performed by the developer while completing a task may be indicative of the effort they are making. The basic actions that a developer may make are typing, moving the pointer, and scrolling. The typing behavior may provide important clues: for example, typing incessantly may indicate that the developer is not spending enough time at checking the code or at thinking at the solution. While moving the mouse and scrolling do not result in production of source code, they may provide interesting pieces of information as well: for example, scrolling continuously when looking at the code may indicate that the developer is struggling at understanding it.

Psycho-Physical Factors

Constraints on time, quality and cost introduce time pressure, stress and emotional trauma in the work field [88, 87, 152, 194]. Physiological aspects could

help in detecting a particular state in which the programmer is in, which may impair the completion a task. The psycho-physical state may both depend on the task and external factors (*e.g.,* the developer's private life or the work conditions): developers may feel frustrated if they cannot understand a specific piece of code or they may be stressed because an important deadline is approaching. Such kinds of conditions may have an impact on the performance. We identified three psycho-physical factors: *stress*, *affective states*, and *concentration*.

Stress can be broadly defined as *an actual or anticipated disruption of homeostasis or an anticipated threat to well-being* [229]. In 1993, the authors of one of the first works in the literature on stress in Software Engineering [236] propose a new perspective on how this factor may affect the behavioral dynamics in the information system development. They state that the stress - with high probabilities - is one of the main corrosive factors for a software developer. Stressor-related information from all major sensory systems is conveyed to the brain, which recruits neural and neuroendocrine systems (effectors) to minimize the net cost. The physiological response to stress involves an efficient and highly conserved set of interlocking systems and aims to maintain physiologic integrity even in the most demanding of circumstances. The autonomic nervous system provides the most immediate response to stressor exposure — through its sympathetic and parasympathetic arms, which provoke rapid alterations in physiological states through neural innervation of end organs. Stress can be detected from biomedical low-invasive sensors. Among others, heart-related measures [123, 233] or their temporal trend [204] were leveraged. It is generally accepted that the activities of the Autonomic Nervous System, which consists of the Sympathetic (SNS) and Parasympathetic nervous systems (PNS), are reflected in the low-frequency (LF) and high-frequency (HF) bands in heart rate variability (HRV) [233]. Some studies support the use of several different physiological factors: Vinkers *et al.* [232] use the body temperature to detect stress, while Suess *et al.* [222] observed that an anomalous respiratory activity (*e.g.,* hyperventilation) may indicate stress as well. Finally, changes in the electrodermal activity due to sweat production is also leveraged for the non-invasive assessment of stress of many kinds [189].

Affective states refer to the emotional response triggered by an internal (e.g. feeling of failure) or external event (e.g. peers ask for help). For representing the

fuzziness of this emotional response, literature adopts the Russell's Circumplex model [200], which describes emotions through the valence (pleasantness of the stimuli) and arousal (level of activation) dimensions. Unpleasant stimuli bring developers to unhappiness, causing the production of software with lower quality than expected [86]. High arousal,instead, could by a symptom of time pressure. When it is not taken under control, it negatively impacts the quality and the efficiency of the development [132].

Concentration can be intended as a set of factors that can measure the ability to think carefully about something the developers are doing and nothing else. For example, in the work proposed by Ahrens *et al.* [3] the focus is given to the attention factor. Indeed, the authors—motivated by the fact that navigating through code and searching for relevant information requires a lot of developer time—designed an approach to use eye tracking to record and transfer developers' attention during software maintenance. The output is offered to the developers in form of a heat map of attention levels. Results showed that that this attention representations helped some of the participants for orientation and code finding purposes, but the majority rated them as barely helpful or even not helpful. Focus is another aspect that can be related to concentration. It was analysed in the work proposed by Soto *et al.* [217], where the authors defined an experiment involving with 14 professional knowledge workers in their workplace over an eight-week with a large variety of biometric measurements (e.g., Heart Rate, Skin Temperature, Respiration Rate). The results showed that the focus can be predicted with an overall accuracy equal to 67%. The concept of staying focused was also reported in the work by Pilzer *et al.* [187] where the authors conducted a formative study with 18 professionals in which they examined their computer-based and eye-gaze interaction with the window environment and devised a relevance model of open windows. The results showed that the devised model was able to predict the relevance of open windows with an accuracy of 72.7%. Finally, interruptibility can be a factor related to concentration. Indeed, according to Zuger *et al.* [254], knowing a person's interruptibility allows optimizing the timing of interruptions and minimize disruption. The authors conducted a two-week field study with 13 professional software developers to investigate a variety of computer interactions such as heart-, sleep-, and physical activity-related data. Their analysis showed

that computer interaction data is more accurate in predicting interruptibility at the computer than biometric data (74.8% vs. 68.3% accuracy), and that combining both yields the best results (75.7% accuracy).

Contextual Factors

The performance of a developer may depend on aspects related to the developer's background and the context in which the task is performed. Specifically, we focus on the following aspects:

- **Developers' experience**: all else being equal, novice developers may introduce bugs more often than expert developers. This could not also be always true because there are many factors that may contribute to the production of source code. What is often the case however is that novice developers take more time to complete the given task. Therefore, considering the protocol of our experiment, the same time is given to every participant. This could provoke that less experienced developers may introduce more bugs;

- **Task Time**: the time at which a task is performed may influence the performance: some developers may work better in the afternoon (or, even, in the night), while others may prefer working in the morning.

7.2.2 Candidate Predictors

We previously introduced the aspects that may affect the developers' performance during a task. We now describe in details the predictors we measure to capture the abstract factors we defined.

Our approach requires to use several different tools and measurement devices. First, it is necessary to use a tool for capturing the users' behavior while developing (*i.e.,* keystrokes, mouse movements, active window at a given time). We use the software *activity-tracker* available online[3] that is a simple tool to log the activity of the user in terms of many measures, such as the number of types, the mouse movements, the time spent on the browser, the activity on the browser.

[3]https://github.com/HASE-UZH/PersonalAnalytics

To capture psycho-physical aspect, it is necessary to use two devices that capture signals from the developers while they are performing the task. In this context, we use a device that captures the heart- and skin-related activity, *i.e.,* Empatica, and a device that captures brain-related activity, *i.e.,* BrainCo. Empatica is a medical-grade wearable device (in the form of a wrist worn bracelet) that offers real-time physiological data acquisition[4]. BrainCo is a single frontal electrode for the acquisition of the EEG and related aggregated data, such as attention and meditation[5]. It is worth noting that some of the predictors we use to capture the aspects we previously explained strongly rely on the measurements provided by such tools/devices.

Behavioral Features

To capture the high-level *activity* performed by the developer, we measure the time that each window receives focus. Then, we use some simple rules to determine the type of activity performed by the developer. The tracker tool we use records the time intervals in which each window received focus and the process that is running behind the window. We focus on two main categories of activities: browsing the web and working in the IDE. Therefore, we define three features that capture the *usage context* (UC):

- **Browser Time (UC_b)**: the percentage of time spent in browser windows;
- **IDE Time (UC_i)**: the percentage of time spent in the IDE;
- **Other Time (UC_o)**: the percentage of time spent in other windows, such as the desktop.

To capture the low-level *actions* made by the developer, we keep into account all the interactions that a user may have with the computer. Specifically, we use the following features:

- **Global Number of keystrokes (KS)**: we simply measure the number of keys pressed by the developer during the whole task;

[4]https://www.empatica.com/en-eu/research/e4/
[5]https://www.brainco.tech/

- **Contextual Number of keystrokes ($\mathbf{KS}_c$)**: we measure the number of keys pressed by the developer in three different contexts: (i) while using the IDE (KS_i), (ii) while using a web broswer (KS_b), and (iii) while using other programs (KS_o).
- **Mouse Cursor Movement (MCM)**: we measure the length of the path traveled by the mouse cursor during the task (in pixels);
- **Mouse Clicks (MC)**: we record the number of clicks made by the developer during the task;
- **Mouse Scroll (MS)**: we measure the number of pixels scrolled by the developer in any direction during the task.

Psycho-Physical Features

In most of the cases, we capture the psycho-physical aspects previously introduced with some signals from the two devices we use, *i.e.,* Empatica and BrainCo. First, we need the signal s in a period in which the developer is completely relaxed: this allows us to calibrate the features for the specific person and to avoid that specific physiological conditions (*e.g.,* a naturally slower HR) make the feature only valid for a single developer or a specific category of developers. Given a signal s obtained during the task and the same signal s_r obtained while relaxing, we first remove outlier samples, possibly due to movement artifacts, by using a robust outlier detection technique [199]. Then, we compute two types of features. First, we extract generic descriptive statistics: we compute the mean and the variance of the signals: we do this to give the machine learning model a rough idea of the state of the developer. If a developer is particularly stressed on a specific day, their average heart rate may be higher than usual, on average, during the whole task. Then, we compute 10 features that describe the relative distribution of the signals: given a signal that ranges between a minimum value m and a maximum value M, we divide the range $[m, M]$ in 10 equally-sized bins and, for each of them, we compute the percentage of the samples of the signals that fall in the specific bin. This allows the model to have an idea of the distribution of the signal. For example, if the signal ranges between 1 and 10, the first

bin (b_1) will contain samples in the range $[1, 2)$, b_2 will be related to the range $[2, 3)$, and so on. We extract the features previously described in two separate contexts: (i) for the whole task, and (ii) just for the last minute of activity. The data about the whole task of the signal gives the model a broad idea of the shape of the signal. On the other hand, the data about the last minute could help getting more interesting information: such data may help capturing a state of relief or stress at the end of the task.

To capture the *stress* level, we keep into account the following aspects:

- **Heart Rate (HR)**: this simple measure allows us to have a rough idea of the psycho-physical state of the developer; a high HR possibly indicate anxiety, while a low HR indicates calm. Given the HR signal, we extract the features for the whole task, *i.e.,* HR_{mean}, HR_{var}, and HR_j (for j between 1 and 10), and the features for the last minute, *i.e.,* HR^{last}_{mean}, HR^{last}_{var}, and HR^{last}_j (for j between 1 and 10) as previously described.

- **Electrodermal Activity (EDA)**: this measure—together with heart related information— can be a reliable indicator of the stress level [2]. Given the EDA signal, we divide it in phasic (EDAP) and tonic (EDAT) using the algorithm proposed by Greco *et al.* [89]. Therefore, given such two signals, we extract the features for the whole task, *i.e.,* $\text{EDAP}_{mean}/\text{EDAT}_{mean}$ $\text{EDAP}_{var}/\text{EDAT}_{var}$, $\text{EDAP}_j/\text{EDAT}_j$ (for j between 1 and 10), and the features for the last minute, *i.e.,* $\text{EDAP}^{last}_{mean}/\text{EDAT}^{last}_{mean}$ $\text{EDAP}^{last}_{var}/\text{EDAT}^{last}_{var}$, $\text{EDAP}^{last}_j/\text{EDAT}^{last}_j$ (for j between 1 and 10). To estimate the number of times in which the stress starts to increase, we introduce an additional feature that extracts the number of times that the first derivative of the signal is positive and greater than an ϵ which allows to be tolerant to small errors by the sensor. We compute such a feature for both the signals, both for the whole task (EDAP_{incr} and EDAT_{incr}) and for the last minute ($\text{EDAP}^{last}_{incr}$ and $\text{EDAT}^{last}_{incr}$). We use $\epsilon = 0.01$.

- **Interbeat Intervals (IBI)**: the heart activity is an indicator of stressful situations. Therefore, we used also a finer observation of Heart Rate Variabilitty (HRV), such as the IBI information. The variation between successive heartbeats is low whether the system of a human is in more of a

fight-or-flight state. The variation between beats is high if one is in a more comfortable state [28]. In particular, we used aggregated data such as the mean and the variance [90].

- **Blood Volume Pressure (BVP)**: another means of stress measurement is made available by way of [90]. For this reason, we involved this measure—available from the smartwatch—in our analysis.

While such features are interesting to capture on their own, they may influence more or less the actual result depending on how hard the developer was working while in a stressed state. For example, a state of stress occurring while trying to understand the problem may impact less the end result than a state of stress occurring while writing code. Therefore, we computed a set of features that weights keystrokes by the stress-related signals previously described. We call such features **Weighted keystrokes (WKS$_c$)**: for each typing session (period of time in which the developer types consecutive characters) we measure the stress-related signals, we normalize and average them, and we use the resulting value as a weight for the number of keystrokes typed in that period. Finally, we sum all such weighted keystrokes to obtain the final WKS$_c$ value. We do this using the signals related to heart rate (WKS$_{HR}$), electrodermal activity (WKS$_{EDAT}$ and WKS$_{EDAP}$), interbeat intervals (WKS$_{IBI}$), and blood-volume pulse (WKS$_{BVP}$).

To capture the *concentration* level, we rely on two factors:

- **Attention level (ATT)**: we use an attention measure provided by the BrainCo device. Such a device uses a proprietary algorithm to combine several brain wave signals to determine the attention level of the user at a given time. Such a measure ranges between 0 (completely relaxed) and 100 (very concentrated). Also for such a signal, we compute the metrics as previously described. Specifically, we compute: ATT$_{mean}$, ATT$_{var}$, ATT$_j$ (for j between 1 and 10), ATT$^{last}_{mean}$, ATT$^{last}_{var}$, ATT$^{last}_j$ (for j between 1 and 10).

- **Tiredness (TIR)**: we ask the developers to self-assess their tiredness on a Likert scale from 1 (not tired) to 9 (completely tired) both before and after

the task. Therefore, we use two features based on such a self-assessment: TIR_{before} and TIR_{after}. The presence of both the features allows the model to implicitly estimate also to what extent the task tired the developer. We also use a binary feature, TIR_{first}, that indicates whether the task was the first of the day (*true*) or the developer completed another complex task before the task at hand (*false*).

Finally, we use the SAM [25] model[6] to capture the *emotional state* of the developers. We do this both *before* and *after* the task. SAM provides that each characteristic of the PAD model (pleasure, arousal, and dominance) is represented through the use of a graphic character arranged along a discrete scale. As for the pleasure, the SAM starts from an initial configuration formed by a smiling and happy figure up to a frowning and unhappy figure. As for the excitement, instead, the SAM starts from a figure of a sleepy character with closed eyes until the excitement, represented by the same character with open eyes. The dominance scale shows the SAM ranging from a very small figure — which aims to represent a feeling not having the situation under control — to a very large figure, which represents a feeling of control or power. We use three features measured on a 9-point Likert scale [162].

Contextual Features

There are several ways in which *developers' experience* can be captured. In this case, we focused on the programming experience and we asked the developers to report the number of years of programming experience in two contexts: *general programming experience* (regardless of the programming language) (PE_{gen}), and *specific programming experience* (on the specific programming language used to complete the task) (PE_{spec}).

Finally, to capture the time of the day at which a task is completed (*task time*), we use a binary categorical variable which can have value *AM* (between 7AM to 12PM) or *PM* (between 12PM and 7PM), depending on when the task was completed.

[6]The Self-Assessment Manikin (SAM) is a technique of non-verbal pictorial evaluation that specifically tests the enjoyment, excitement, and dominance associated with the affective response of a person to a wide range of stimuli.

Combined Features

We opted for the creation of a new set of features with the aim at increasing the knowledge of the data set and improving the classification performances. The new set of features has been defined by weighting the number of typed keys by all the features measured using Empatica device (including the temperature). Consider the number of keys $K_{s,e}$ typed in a time slot $[s, e]$. Consider, than, a given Empatica feature $f_{s,e}$ computed in the same time interval. We compute the number of keystrokes weighted by f using the following formula:

$$K^f = \sum_{(s,e) \in KS}^{n-1} K_{s,e} norm(f_{s,e})$$

where KS contains the couples of start and end time s and e for which the key tracker recorded the number of keystrokes, and *norm* is a function that normalizes the f feature between 0 and 1 by simply using the formula $norm(x) = \frac{x-min}{max-min}$. The features f we consider are all the ones recorded by Empatica, *i.e.,* the body temperature T, the Blood Volume Pulse BVP, the Interbeat Interval IBI, the Hearth Rate HR, and the two EDA features (*i.e.,* phasic and tonic). We also included three additional features in which we measure the number of keystrokes divided by the environment in which the participant was (IDE, web browser, or other).

CHAPTER 8

Evaluation of the Framework

Contents

8.1 Controlled Experiment

Our study is steered by the following research questions:

RQ_1 *Is it possible to predict the outcome of a task using the task-based features we introduced?* With this research question we try to use the features we previously listed to predict if the developer will introduce at least a bug in the code; we compare our model with a classical bug prediction model that uses code-related features [170];

RQ_2 *Is a combined code- and developer-based model more effective than separate models?* With this research question we want to understand if a model obtained combining our new features with the code-related features of the baseline used to answer RQ_1 allows to achieve a higher classification accuracy.

To answer our research questions, we conducted a controlled experiment in which we invited 22 developers to complete a total of 4 tasks while we could track their activity and their psycho-physical state using the devices we previously described.

8.1.1 Experiment Design

Our study includes both *subjects* (human developers) and *objects* (tasks to be performed). As for the subjects, we involved a group of software developers composed of both students at the University of Molise (PhD candidates, master and bachelor students) and professional developer. In total, we involved 22 developers. We only involved bachelor students that passed the Java exam provided at the first year of the Program in Computer Science at the University of Molise. To avoid involving young developers with not enough experience, we used an additional requirement for second-year students (the youngest involved in our study), *i.e.,* only selected the ones that passed the exam with the highest score (30/30). We had to discard two of the invited developers since, because of a memory problem of one of the devices, we lost part of the signals we acquired for them.

We report in Figure 8.1 a summary of the demographics of the participants. More than half of them (13) is composed by young developers (bachelor students at the 2^{nd} or 3^{rd} year), while 5 are Master Students, 3 are PhD students, and 1 is working in industry. In addition, we investigated the experience in programming.

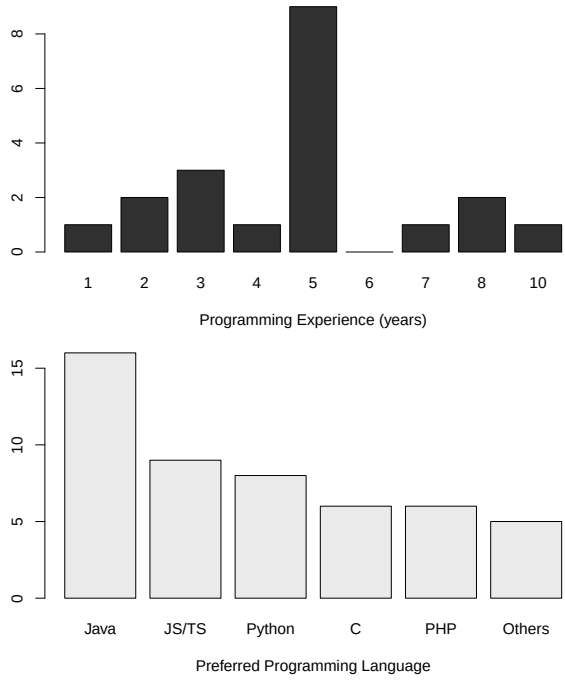

Figure 8.1: Descriptive statistics about the participants involved in the experiment. The bar plot at the top reports the generic programming experience (in years), while the one at the bottom depicts the programming languages with which the participants declared to feel more confident.

The distributions are depicted in the top right of the bar chart from Figure 8.1: 36.4% of the developers have at least 5 years of programming experience. Finally, in the bottom graph of the figure, we report the programming languages in which the developers feel more proficient.

As for the objects, we first selected four problems from *LeetCode*, an online platform commonly used by developers to exercise for coding interviews. Such a platform allows to access a wide range of problems and it provides a mechanism for validating a solution: given the source code of the solution, the platform runs several test cases (depending on the problem) and it reports the number of failed tests. A solution is *accepted* if all the test cases pass. Besides, for each problem,

Table 8.1: Tasks selected from LeetCode for the controlled experiment.

Difficulty	Type	Problem name	Acceptance rate
Hard	Implementation	Roman to Integer	55.1%
Hard	Bug Fixing	Camelcase Matching	56.1%
Easy	Implementation	Split a string in balanced strings	82.4%
Easy	Bug Fixing	Robot return to origin	73.2%

LeetCode reports the acceptance rate, *i.e.,* the percentage of submissions by the community that were accepted/rejected by the platform. We used such a value to select the problems for the experiment. First, we selected two *easy* problems by choosing problems with an acceptance rate greater than 70%. Among the candidate problems, we selected the ones that, according to our evaluation, could be completed in about half an hour. Then, we selected two *hard* problems: we wanted the problems to be harder than the easy problems, but not so hard that no one would have been able to correctly complete them in half an hour. Therefore, we selected two problems with an acceptance rate between 50% and 60%. In the further sections of this thesis this distinction will be categorized in *easy* and *hard* tasks but given the percentages of acceptance rate, it would have been more fair to refer to hard tasks as *medium difficulty* tasks.

Starting from the selected problems, we defined four tasks, two implementation tasks and two bug-fixing tasks. The implementation tasks consisted in solving the problem from scratch. We randomly chose two of the problems selected from LeetCode to define such tasks, one easy and one hard. The bug-fixing tasks required the developers to read a partial solution to the problem that contained at least a bug and to fix all the bugs. Two of the authors developed a partial solution for the two remaining problems. We report in Table 8.1 the list of the tasks we used.

In the design of our experiment, we controlled the following variables:

- **Task type**: as previously mentioned, we asked the developers to complete two bug-fixing tasks and two implementation tasks;
- **Task difficulty**: each developer was asked to complete two easy and two hard tasks;

- **Time of the day**: we made sure that each developer performed exactly two tasks in the morning and two tasks in the afternoon;

Because of this design, we defined eight groups, and we divided the developers in such groups to avoid that any of the previously described variables influenced the results. While dividing the developers in groups, we balanced them based on their education level (*i.e.,* we avoided groups with all Master students and groups with all first-year Bachelor students). To avoid that tiredness negatively affected the task performance, we the developers completed two tasks in a session and two tasks in another session (on different day). The participants were asked to complete the tasks by using the Java programming language since they all had experience with it. They were provided with a laptop with IntelliJ IDEA Community Edition and the software needed for tracking their activities already installed. Also, they were asked to wear the two devices needed to capture their psycho-physical-related signals (*i.e.,* Empatica and BrainCo). We submitted questionnaires to the participants both before/after the whole experiment and before/after each task. Finally, since some psycho-physical metrics required measurements during a relax period, we asked the developers to watch a relaxing video for two minutes before starting the task to acquire such signals.

8.1.2 Data Collection

We collected (i) the biomedical signal data, (ii) the activity tracker recordings, (iii) the contextual information, and (iv) responses given by the developers to the questionnaires to compute the features we defined in the previous section. To define the outcome of the task, *i.e.,* to label each instance as *correct* or *buggy*, we used LeetCode itself: we tried each solution to each problem, and we checked if the platform reported any bugs. If it did (*i.e.,* at least a test case failed), we marked the instance as *buggy*, while we labeled it *correct* otherwise.

8.1.3 Data Analysis

In our experiment, for the feature selection technique, we decided to use a Wrapper approach [126]: we used Linear Forward Selection [91] to select candidate sets of features and, to evaluate each set, we trained and tested a logistic

regression classifier. We used the AUC as the metrics to optimize. It is worth noting that we used different strategies. Therefore, to answer our research questions, we simulated two different scenarios:

- *Newcomer Scenario*: *newcomers* represent developers in their first day in a development group. With this scenario, we wanted to validate our approach when no personal data is available in the training process of the model. To reproduce this scenario, we have adopted a classical *Leave One Subject Out* Cross Validation (LOSO-CV). The data has been decomposed in n folds, one for each subject. Then, we used each of such folds — composed of all the four instances of a given subject — as test set and the union of the remaining folds as training set. As a result, the data related to a single subject appears once in the test set and *n-1* times in the training set. It is worth noting that this is a challenging scenario since the machine learning technique can not learn from any peculiarities of a specific subject.

- *Freshman Scenario*: *freshman* are developers who recently joined a software development group, but they are not on their first work day; in this scenario, our aim was to evaluate the proposed approach when *few* data are available for a given subject. To simulate such a scenario, we used a *Leave One Task Out* Cross Validation (LOTO-CV) or, simply, Leave One Out, since a task is an instance, in our context. We used three of the tasks completed by a given developer and all the tasks completed by the other developers as training set and the remaining task by the developer as test set.

In the Newcomer Scenario, we perform feature selection for each model (*i.e.,* the model built for each participant has its own features): we do this because we completely exclude from the training data set any information related to the participant. In the Freshman Scenario, instead, we perform feature selection only once (*i.e.,* all the models have the same features). Finally, we used the Multilayer Perceptron [177] to build our classifier. We used the implementations of all the mentioned algorithms available in the Weka [93] toolkit.

To answer RQ_1, we compare our approach with a state-of-the-art approach. We could choose among many possible bug prediction approaches available in the literature. The most recent approaches, however, are designed to work for big

Object-Oriented programs for which a revision history is available. In our case we don't have such an information. For this reason, we could not use process metrics and we could only consider product metrics. Also, we had to exclude most of the CK metrics since we have at most three classes in a program, while most of them are implemented in a single class or even method. We used as the baseline the approach introduced by Nagappan *et al.* [170], who considered a set of metrics that could be computed also at method-level. In detail, we considered the following metrics:

- *Classes*: number of classes
- *Functions*: number of functions
- *Lines*: number of executable lines
- *Parameters*: number of parameters
- *Arcs*: number of arcs in control flow graph
- *Blocks*: number of basic blocks in control flow graph
- *FanIn*: number of calling functions
- *FanOut*: number of functions called
- *Complexity*: McCabe's cyclomatic complexity

To answer RQ_2, we first computed the overlap metrics between our developer-based model and the baseline. To do this, we computed the percentage of instances correctly classified (i) only by using a developer-based model (*OnlyD*), only by using a code-based model (*OnlyC*), by both the models (*Common*). Specifically, given the set of instances correctly classified by the developer-based model, D, ad the set of instances correctly classified by the code-based model, C, we computed the metrics as follows:

- $OnlyD = \frac{|D \setminus C|}{|D \cup C|}$
- $OnlyC = \frac{|C \setminus D|}{|D \cup C|}$
- $Common = \frac{|C \cap D|}{|D \cup C|}$

As a second step, we defined a combined model. To do this, we used all the developer-based (our model) and code-based (baseline, previously described in RQ_1), trained and tested in both the newcomer and freshman scenario, similarly to what we did to answer RQ_1. We compare the results achieved with the results obtained using the single models.

8.2 Analysis of the Results

In this section, we report the results of our study. First, we answer our research questions, and then we discuss some interesting cases we found. We also use individual detectors to check if some of them can be avoided. The full data set used for the analysis of the results is composed of 80 data points, each related to a task performed by a developer. The actual number of tasks is divided into 48 *buggy* and 32 *non-buggy* tasks. Therefore, a *constant* classifier—*i.e.,* a classifier always predicting the majority class—would obtain 48 out of 80 tasks correctly classified with an overall accuracy of 60%. We also did not report the comparison of our model to a random classifier because we experimented it and our model always outperform random classification models (for this purpose, we used a Random Forest[16]).

8.2.1 RQ_1: Effectiveness of a Developer-Based Bug Prediction Model

We describe below the results obtained in the Newcomer Scenario (no previous data points available for the developer) Freshman Scenario (few data points available for the developer).

Newcomer Scenario

The most selected features in the Newcomer Scenario are reported in Table 8.2. It is worth noting that, in this scenario, we build several models and, for each of them, we performed a separate feature selection. This means that we have several models, each with different sets of features. For this reason, we also report the number of folds in which each feature appears, and we rank the

Table 8.2: RQ_1: The results of the features selection phase in the Newcomer Scenario.

Type	**Source**	**#Folds**	**Feature**
Psycho-Physical	Empatica	18	IBI_{mean}
Psycho-Physical	Empatica	16	EDAT_{10}^{last}
Psycho-Physical	Empatica	15	HR_{var}^{last}
Psycho-Physical	Tracker/Empatica	15	WKS_{HR}
Psycho-Physical	Empatica	13	HR_{var}
Psycho-Physical	Empatica	12	EDAP_{10}
Psycho-Physical	Empatica	10	HR_{1}

Table 8.3: RQ_1: Comparison in terms of accuracy between our developer-based model and the baseline code-based model.

Scenario	**Developer-based**	**Code-based**
Newcomer	68.8%	**71.3%**
Freshman	**86.3%**	80.0%

features according to this value. We only report the features selected in at least 10 folds.

The most selected features are related to the heart activity even with the keystroke feature weighted according to the number of keystrokes executed during the tasks.

We report in Table 8.3 the comparison between the two models. The developer-based model achieves 68.8% accuracy, in total. Specifically, the model correctly classified 55 out of 80 tasks as *buggy* or *Non-buggy*, in the newcomer scenario. Of these, 38 out of 48 and 17 out of 32 were correctly classified as *buggy* and *non-buggy* tasks, respectively.The baseline achieves a higher level of accuracy since it correctly classifies 71.3% of the tasks (57 out of 80). We report the detailed results by class for our model in Table 8.4. It can be noticed that our developer-based model has a low precision and, mostly, recall on *Non-buggy* instances. This means that the model reports many false negatives (*i.e.,* it tends to say that task was not correct even if it was).

Table 8.4: RQ_1: Detailed results of our developer-based model.

Scenario	Class	Precision	Recall	F-measure
Newcomer	*Buggy*	0.717	0.792	0.753
	Non-buggy	0.630	0.531	0.576
Freshman	*Buggy*	0.894	0.875	0.884
	Non-buggy	0.818	0.844	0.831

Table 8.5: RQ_1: The results of the features selection phase in the Freshman Scenario.

Type	Source	Feature
Psycho-Physical	Empatica	EDAP_2
Psycho-Physical	Empatica	EDAP_{10}
Psycho-Physical	Empatica	EDAT_{10}^{last}
Psycho-Physical	Empatica	IBI_{mean}
Psycho-Physical	Empatica	HR_{var}
Psycho-Physical	Empatica	HR_1
Psycho-Physical	Empatica	HR_{var}^{last}
Psycho-Physical	Empatica	HR_2^{last}
Psycho-Physical	Empatica	HR_3^{last}
Psycho-Physical	Empatica	HR_{10}^{last}
Behavioral	Tracker	MCM
Psycho-Physical	Tracker/Empatica	WKS_{BVP}
Psycho-Physical	Tracker/Empatica	KS_{IDE}
Psycho-Physical	Tracker/Empatica	$\text{KS}_{Browser}$

Freshman Scenario

We report the features selected in the Freshman Scenario in Table 8.5.

In this case, the nature of the most selected features is the same as in the newcomer scenario. Indeed the heart, EDA and keystroke activity seem again very important indicators of correctness of a task. Our developer-based model achieves 86.25% accuracy: in details, the classifier has correctly classified 69 out of 80 tasks. The code-based model, instead, achieves 80.0% accuracy (64 out of 80 correctly classified tasks). Of these, 42 out of 48 and 27 out of 32 were correctly classified as *buggy* and *non-buggy* tasks, respectively. Also for this scenario, we

Table 8.6: RQ_2: Overlap metrics in the two scenarios.

Scenario	OnlyD	OnlyC	Common
Newcomer	22%	25%	53%
Freshman	19%	13%	68%

report the detailed results of our developer-based model in Table 8.4. While also in this case the precision and recall are lower on the *Non-buggy* class, they are more acceptable (always higher than 0.8). Therefore, also in this case, false negatives are higher than false positives. In most of the application contexts, the most expensive error is represented by false negatives (*i.e.,* the model says that the task was correctly executed, while it was not): this results in the introduction of bugs in the system, which may be hard to find in the future. On the other hand, false positives (*i.e.,* the model classifies correct tasks as *buggy)* are, generally, less expensive and more acceptable: they may result in more in-depth code reviews or other kinds of QA practices.

Summary of RQ_1: A developer-based bug prediction model is able to achieve a higher accuracy compared to a code-based model, but only in a scenario in which some data about the developer are provided in the training set (+6.3% accuracy in the Freshman Scenario).

8.2.2 RQ_2: Effectiveness of a Combined Model

We describe the results obtained in the two scenarios we considered, *i.e.,* Newcomer Scenario (no previous data available for the developer) Freshman Scenario (few data available for the developer).

Newcomer Scenario

We report the overlap metrics in Table 8.6. There is a considerable percentage of instances (about half of them) that can only be correctly classifying considering either a developer-based or a code-based model; indeed, only 53% of the instances can be correctly classified by both the models.

Table 8.7: RQ_2: The results of the features selection phase in the Newcomer Scenario for the combined model.

Type	Source	#Folds	Feature
Psycho/Physical	Empatica	20	EDAT_{10}^{last}
Psycho/Physical	Tracker/Empatica	20	WKS_{HR}
Behavioral	Tracker	20	MCM
	Code	19	LOC
	Code	15	$\#\text{Parameters}_{max}$
	Code	10	$\#\text{Parameters}_{total}$

Table 8.8: RQ_2: Detailed results of the combined model.

Scenario	Class	Precision	Recall	F-Measure
Newcomer	*Buggy*	0.761	0.729	0.745
	Non-buggy	0.618	0.656	0.636
Freshman	*Buggy*	0.933	0.875	0.903
	Non-buggy	0.829	0.906	0.866

As for the combined models we trained and tested in the Newcomer Scenario, the most selected features in this scenario are reported in Table 8.7, ranked by the number of folds. In the table, only features selected at least in 10 folds have been reported.

This time, no individual heart-related predictors were selected. The only one that concerns the heart activity was the number of keystrokes weighted with respect to the heart rate. Furthermore, the EDA Tonic values in distribution 10 were the only biometric parameters considered in this experiment. Finally, behavioral and code-based features were predominant. We report in Table 8.8 the performance of the model. Overall, the combined model achieved 70% accuracy (56 correct classifications out of 80). While the two models are complementary, as the overlap metrics show, simply combining the features does not allow to improve the accuracy: the code-based model alone correctly classifies one additional instance compared to the combined model (57 out of 80).

Table 8.9: RQ_2: The results of the features selection phase in the Freshman Scenario for the combined model.

Type	Source	Feature
Psycho/Physical	Empatica	EDAT_{10}^{last}
Behavioral	Tracker	MCM
Psycho/Physical	Tracker/Empatica	WKS_{HR}
Behavioral	Tracker	KS_{IDE}
	Code	$\#\text{methods}_{max}$
	Code	$\#\text{parameters}_{max}$
	Code	LOC

Freshman Scenario

Also in this case, we report the overlap metrics in Table 8.6. In the Freshman Scenario, the overlap between the two models is higher (68%). This also happens because the performance of both the models in this scenario is higher (*i.e.,* it is more likely that both correctly classify the same instance). It is worth remarking, however, that 19% of the instances can only be correctly classified by using developer-based features.

As for the combined models, the features selected in this scenario are reported in Table 8.9.

In this case, the combined model achieves 88.8% of overall accuracy. In details, the classifier has correctly classified 71 out of 80 tasks. The detailed performances by class achieved by this classifier are described in Table 8.8.

Summary of RQ_2A developer-based bug prediction model is complementary to a code-based model: between 19% (Freshman Scenario) and 22% (Newcomer Scenario) of the instances can be correctly classified only with such a model. A model combining all the features achieves ∼89% accuracy in the Freshman Scenario.

8.2.3 Discussion

A summary of this work is shown in Figure 8.2. The figure also shows the features and the classifier we used in the experiment to reach the best effective-

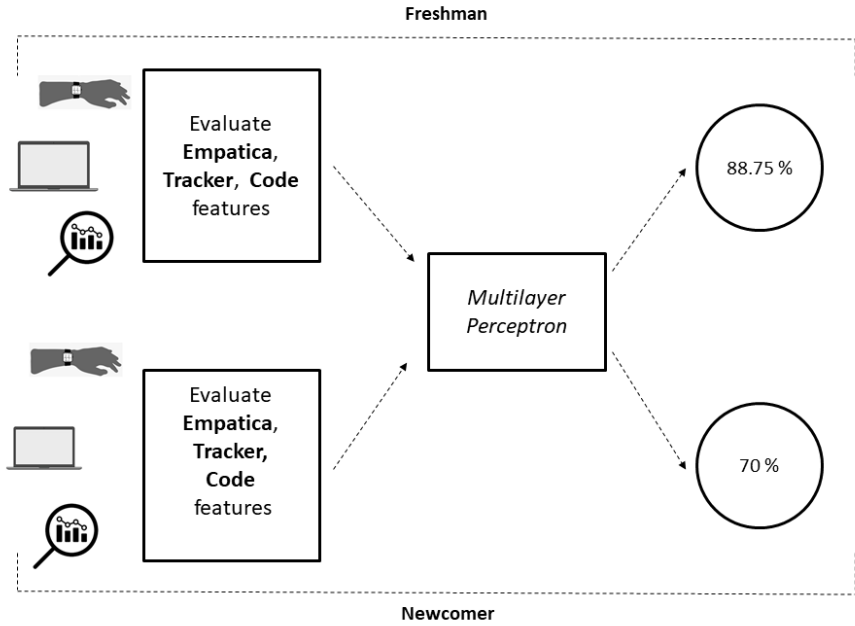

Figure 8.2: Results achieved by the proposed approaches in all the scenarios and for both the classification experiments. The figure also shows the features to evaluate and the classifier to use in order to achieve the best classification performances.

ness. In this section we (i) discuss some examples in which the developer-based model failed to predict the correctness of the task, and (ii) report an additional analysis in which we try to understand which information sources are important and which ones can be avoided.

Examples of Negative Results

We analyzed more in-depth the cases in which the developer-based bug prediction model provided the worst results to understand what did not work and what could be the future research directions. Specifically, we focused on the Newcomer scenario, *i.e.,* the one in which we achieve the worst results, and we

took a closer look at the subject for which the number of correct predictions was lowest.

For the subject for which we obtained the worst results in the Newcomer Scenario (PLEMR, one correct classification out of four), the classifier predicted *Non-buggy* three times, one of which was correct, while it predicted *Buggy* only once, and it was not correct.

Looking at the features, we found that the subject (i) typed less than 1,000 keys in all the task, and he/she shows average values of EDA phasic higher than other participants. We tried to train and test a rule-based classifier (JRip [44]) to understand, specifically, which features contributed to the misclassification based on the defined rules: we found that the $EDAP_{10}$ of the participant was beyond the threshold for classifying it as *Buggy* even when he/she correctly completed the task. This possibly means that the participant was stressed for other reasons, but he/she was able to correctly complete the task anyway. According to the answers to the pre/post task questionnaires, the developer reported a higher level of SAM1 compared to the median of the all the other developer (except for one task). This roughly indicates that he/she was slightly sadder than the other participants. Also, it can be noticed that his/her SAM levels reported before and after the task were always the same (except for a case in which one of the values changed by 1).

Individual Information Sources

Some sensors necessary to compute some of the features are rather uncomfortable to wear in a normal working environment. Also, asking questions to a developer before and after a task may, as well, limit the possible applicability of our approach. In this section, we want to understand which information sources are more important and which ones can be safely avoided, without impacting the model accuracy. We focused only on the scenario for which the model works best, *i.e.,* the Freshman Scenario. We used several different classifiers available in the Weka toolkit [93], covering all the main categories. Specifically, we used:

- BayesNet [24] from the bayes category;

Table 8.10: Evaluation of individual information sources.

Source	Type	Invasiveness	Accuracy
BrainCo	Headband	High	61.25%
Empatica	Wristband	Medium	76.25%
Activity Tracker	Program	None	63.75%
Questions	Questionnaire	Low	57.50%

- SGD[1], Logistic Regression [141] and Multilayer perceptron [161] from the functions category;
- AdaBoostM1 [68] from the meta category;
- JRip [44] from the rule category;
- RandomForest [26] and Decision Stump[2] from the tree category.

Table 8.10 reports a summary of the accuracy achieved by each individual detector. We discuss below each of them.

BrainCo (Headband). The selected features are related to the attention level (ATT_1, ATT_7, ATT_8, ATT_7^{last}) The best classifier — in terms of overall accuracy — is MultiLayer Perceptron, with 61.25% accuracy. We found that wearing a headband is not comfortable to developers: setting up the sensor requires some experience, to ensure that the signal is acquired correctly, and it exerts pressure on the developers' head. Also given the rather small contribution of the features provided by the headband, we do not recommend using such sensors.

Empatica (Wristband) The selected features are related to the EDA ($EDAP_6$, $EDAP_8$, $EDAP_{10}$, $EDAT_{10}^{last}$), IBI (IBI_{mean}), and HR (HR_1, HR_{var}^{last}, HR_2^{last}, HR_7^{last}). The best classifier — in terms of overall accuracy — is the Logistic with 76.25% accuracy. We found that the wristband was quite easy to setup since no experience was required. While some developers indicated that the wristband was tight (necessary to better acquire the signal) and, therefore, a little unconfortable, we registered no major complaints. Given the high impact that the

[1]https://bit.ly/2K8t0Xx
[2]https://bit.ly/2Wh2FsU

features computed from the signals acquired through such a sensor, we highly recommend using this kind of sensor.

Activity Tracker. The selected features are MCM (mouse cursor movement) and KS (keystrokes). The best classifier — in terms of overall accuracy — is the SGD with 63.75% accuracy. Since the activity tracker is a program that runs in background, this does not bother at all the developers. Even if such features, alone, do not allow to accurately predict the correctness of the task, we found them appearing many times in the final models investigated in RQ_1. For these reasons, we recommend using this source of information.

Questionnaire. The selected features are PE_{spec} and SAM_{post}. The best classifier — in terms of overall accuracy — is the Random Forest with 57.5% accuracy. Answering few questions before/after a task may be a little bothering during the development activities; however, this requires less than a minute. Besides, the questions could be integrated in the IDE through a plugin, making the acquisition of such information even less invasive. However, the accuracy achieved by such features, alone, is very low and they are never selected in the models (RQ_1 and RQ_2). Based on the context, we cannot give a definitive recommendation: we believe that the final decision mostly depend on the application context.

Threats to Validity

Most of the threats to the validity for our results are related to the external validity. The results may be mostly valid for our sample of participants and the specific tasks we chose. We tried to minimize this threat by (i) involving both more and less experience junior developers, and (ii) choosing both easy and difficult programming tasks. Moreover, it is worth noting that our results only refer to relatively small programs and short time spans.

The biggest threat to the internal validity of our study is the monitoring of participants: this might have implicitly influenced their behaviors since they could feel observed. We minimized this threat by using between-group design to avoid learning effects, assigning the same amount of tasks from set A and B to both groups and giving all instructions in written form. Also, some biomedical metrics are influenced by the individual, as well as environmental factors, such

as lighting. We counteracted these confounding factors by having a controlled setup with fixed screen brightness, room temperature, and lighting.

8.3 Final Remarks

We report below the main lessons learned from our study.

Lesson 1: Do not use a developer-based model with newcomers. One of the most important outcomes of our experiment is that the best effectiveness in terms of overall accuracy can be reached when some data of the developer who performs the task are available. Indeed, a developer-based model does not work very well in a Newcomer Scenario, *i.e.,* when it is trained using other developers' data: a simpler code-based model is more effective, in such a scenario. We recommend monitoring for a while a new developer before using such a model with them.

Lesson 2: There is no need for an EEG device. Another important outcome derives from the measurements involved in our experiments. Our results show that the best effectiveness can be obtained by using only a wristband device for monitoring psycho-physical aspects (*e.g.,* the heart-rate), an activity tracking software, and assessing contextual features through a questionnaire. This mean that there is no need of a headband for predicting defects in the source code: we noticed that such a device was also the less comfortable to wear to developers. Or at least, that a single electrode prototype is not able to capture the brain dynamics involved while producing source code.

Part IV

Conclusion

Biomedical monitoring is used in a large variety of applications. Sport is a context where this type of monitoring is becoming day-by-day more crucial and important. Trainers want to know every detail about their athletes' state of health during a training session.

The first contribution of this thesis are contextualized in this scenario. We ideated and defined a set of metrics aimed at objectively measuring a performance in terms of body and test derived information. This has been done for a particular mental test—the stroop test—and it is nowadays used in the industrial software of our partner.

After this, we approached an innovative study dedicated to the definition of a Trainer Support Tool in a simulated scenario. Indeed, our partner AOTech has designed and built a race simulator used by several professional teams. Therefore, we have worked at the integration of the simulated vehicle's telemetry with the body's information (biometry), in order to create a software tool able at helping the trainer in the contextualized monitoring of vehicle and body data and in the process of decision-making.

We have then tried to extend the monitoring of motorsport drivers with studies dedicated to the early diagnosis of pathological cardiac conditions.

For this purpose, we designed several Machine Learning approaches for (i) Atrial Fibrillation episodes, (ii) Premature Ventricular Contractions, (iii) Bundle Branch Blocks, (iv) Atrial Premature Beats identification. We also investigated the possibility to automatically identify anomalies in a compressed ECG. In this particular field of research, we defined two approaches: the first dedicated to the estimation of heart rate and the second involved in the classification of compressed heart beat according to a well-known literature standard.

Finally, with the expertise gained in such contexts, we designed a study for the monitoring of software developer. We believe that factors as concentration and scenario as a free-mind may impact both on contexts of motorsport and production of source code. Considering this parallel between the two contexts, we dedicated part of our research effort to the monitoring of developers while they are executing coding tasks, such as implementation of a new feature or bug-fixing. Our approach is dedicated to the automatic detection of the correctness of a task, *e.g.,* if a task was completed successfully (no bugs) or not. Our results show that

a developer-based model is able at identify the correctness of a task with good accuracy. In addition, the main lessons learned from our study are that (i) a developer-based model is only effective when personal data are available, and (ii) it is possible to remove some of the information sources (*e.g.,* the headband) to reduce the invasiveness of the measurements and still achieve similar accuracy levels.

The main contributions of this work are:

- design and definition of a set of metrics dedicated to provide an objective measure of a mental performance.
- implementation of a software tool to support motorsport trainers and drivers during simulated races.
- design and definition of several automatic detectors of pathological cardiac conditions to be intended as support to specialized medical staff.
- design of a study that involved computer science students and professional developer with the aim at implementing a tool able at predicting the correctness of a coding task, given monitoring data.

Future works will be dedicated to the:

- the follow-up for our controlled experiments may meet the following directions: (i) involve more senior participants, (ii) to corroborate the interesting finding that EEG data can be ignored for defect prediction, (iii) embed better (finer grained) behavioral features. This latter point, for example, can be useful because distinguishing only between IDE time and browsing time, neglects important information from the browser: with a simple browser extension it is possible to distinguish between productive work (browsing IDE documentation, stack overflow, etc.) and distractions (social networks, news, etc.).
- monitoring the biomedical data of developer while they provide code under stress conditions. Monitoring biomedical data in this context would be very challenging, but perhaps the scenario can be simulated.

- extension of automatic detectors of pathological conditions retrievable from an ECG trace.
- applicability of Machine Learning in the compressed domain with the aim at performing more accurate automatic analysis.
- extend the biometric monitoring in other contexts, such as the home rehabilitation.

Appendices

APPENDIX A

Publications

J1 **G. Laudato**, S. Scalabrino, D. Girardi, N. Novielli, F. La Nubile e R. Oliveto "A developer-based bug prediction model." - IEEE Transactions on Software Engineering **(under revision)**.

J2 **G. Laudato**, F. Boldi, A. Colavita, G. Rosa, S. Scalabrino, A. Lazich, R. Oliveto, "Combining rhythmic and morphological ecg features for automatic detection of atrial fibrillation: Local and global prediction models." - Biomedical Engineering Systems and Technologies (Springer) *(under revision)*.

C1 **G. Laudato**, R. Oliveto, S. Scalabrino, A. R. Colavita, L. De Vito, F. Picariello, and I. Tudosa, "Identification of R-peak occurrences in compressed ecg signals." - 2020 IEEE International Symposium on Medical Measurements and Applications (MeMeA). IEEE, 2020, pp. 1–6.

C2 **G. Laudato**, F. Boldi, A. R. Colavita, G. Rosa, S. Scalabrino, P. Torchitti, A. Lazich, R. Oliveto "Combining Rhythmic and Morphological ECG Features for Automatic Detection of Atrial Fibrillation." - 13th International Conference on Health Informatics, HEALTHINF 2020, La Valletta (Malta).

C3 **G. Laudato**, F. Picariello, S. Scalabrino, I. Tudosa, L. D. Vito, and R. Oliveto "Morphological classification of heartbeats in compressed ecg" - 14th International Conference on Health Informatics, HEALTHINF 2021 (*Accepted, to appear*).

C4 G. Rosa, **G. Laudato**, A. Colavita, S. Scalabrino, and R. Oliveto, "Automatic real-time beat-to-beat detection of arrhythmia conditions" - 14thInternational Conference on Health Informatics, HEALTHINF 2021 (*Accepted, to appear*).

BC1 **G. Laudato**, G. Rosa, G. Capobianco, A. Colavita, A. Del Forno, F. Divino,G. Ferraro, C. Lupi, R. Pareschi, S. Ricciardi, L. Romagnoli, S. Scalabrino,C. Tomassini, and R. Oliveto, "Simulating the doctor's behaviour: Identification of atrial fibrillation through combined analysis of heart rate and beat morphology." - DataScienceBook-2020, Data Science and Big Data Analyticsin Smart Environments (*Accepted, to appear*).

A.1 Other Publications

J3 **G. Laudato**, S. Scalabrino, A. Colavita, Q. Chiacchiari, R. D'Orazio, R. Donadelli, L. De Vito, F. Picariello, I. Tudosa, R. Malatesta, L. Gallo, R. Oliveto "ATTICUS: Ambient-intelligent Tele-monitoring and Telemetry for Incepting and Catering overhUman Sustainability" - Frontiers in Human Dynamics, section Digital Impacts *(under revision)*.

J4 M. Antico, N. Balletti, **G. Laudato**, A. Lazich, M. Notarantonio, R. Oliveto, S. Ricciardi, S. Scalabrino, J. Simeone "Postural Control Assessment through MicrosoftAzure Kinect: An Evaluation Study" - Sensors *(under revision)*.

C5 **G. Laudato**, G. Rosa, S. Scalabrino, J. Simeone, F. Picariello, I. Tudosa, L. De Vito, F. Boldi, P. Torchitti, R. Ceccarelli, F. Picariello, L. Torricelli, A. Lazich, Rocco Oliveto "MIPHAS: Military Performances and Health Analysis System" - 13th International Conference on Health Informatics, HEALTHINF 2020, La Valletta (Malta).

C6 E. Balestrieri, F. Boldi, A.R. Colavita, L. De Vito, **G. Laudato**, R. Oliveto, F. Picariello, S. Rivaldi, S. Scalabrino, P. Torchitti, I. Tudosa "The architecture of an innovative smart T-shirt based on the Internet of Medical Things paradigm." - Proceedings of 2019 IEEE International Symposium on Medical Measurements and Applications (MEMEA), Istanbul, Turkey - June 2019

Bibliography

[1] N. Adochiei, V. David, F. Adochiei, and I. Tudosa. Ecg waves and features extraction using wavelet multi-resolution analysis. In *2011 E-Health and Bioengineering Conference (EHB)*, pages 1–4. IEEE, 2011.

[2] A. Affanni. Wireless sensors system for stress detection by means of ecg and eda acquisition. *Sensors*, 20(7):2026. Multidisciplinary Digital Publishing Institute, 2020.

[3] M. Ahrens, K. Schneider, and M. Busch. Attention in software maintenance: an eye tracking study. In *2019 IEEE/ACM 6th International Workshop on Eye Movements in Programming (EMIP)*, pages 2–9. IEEE, 2019.

[4] L. Aleksovska-Stojkovska and S. Loskovska. Data mining in clinical decision support systems. In *Recent progress in data engineering and internet technology*, pages 287–293. Springer, 2013.

[5] Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms. Standard, Association for the Advancement of Medical Instrumentation, Arlington, VA, 1998.

[6] Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms. Standard, Association for the Advancement of Medical Instrumentation, Arlington, VA, 1998.

[7] S. Asgari, A. Mehrnia, and M. Moussavi. Automatic detection of atrial fibrillation using stationary wavelet transform and support vector machine. *Computers in biology and medicine*, 60:132–142. Elsevier, 2015.

[8] J. M. Atkins, S. J. Leshin, G. Blomqvist, and C. B. Mullins. Ventricular conduction blocks and sudden death in acute myocardial infarction: potential indications for pacing. *New England Journal of Medicine*, 288(6):281–284. Mass Medical Soc, 1973.

[9] J. Bai, L. Mao, H. Chen, Y. Sun, Q. Li, and R. Zhang. A new automatic detection method for bundle branch block using ecgs. In *International Conference on Health Information Science*, pages 168–180. Springer, 2019.

[10] S. Baldasseroni, C. Opasich, M. Gorini, D. Lucci, N. Marchionni, M. Marini, C. Campana, G. Perini, A. Deorsola, G. Masotti, et al. Left bundle-branch block is associated with increased 1-year sudden and total mortality rate in 5517 outpatients with congestive heart failure: a report from the italian network on congestive heart failure. *American heart journal*, 143(3):398–405. Elsevier, 2002.

[11] E. Balestrieri, F. Boldi, A. R. Colavita, L. De Vito, G. Laudato, R. Oliveto, F. Picariello, S. Rivaldi, S. Scalabrino, P. Torchitti, et al. The architecture of an innovative smart t-shirt based on the internet of medical things paradigm. In *2019 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 1–6. IEEE, 2019.

[12] E. Balestrieri, F. Boldi, A. R. Colavita, L. De Vito, G. Laudato, R. Oliveto, F. Picariello, S. Rivaldi, S. Scalabrino, P. Torchitti, et al. The architecture of an innovative smart t-shirt based on the internet of medical things paradigm. In *2019 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 1–6. IEEE, 2019.

[13] E. Balestrieri, P. Daponte, L. De Vito, F. Picariello, S. Rapuano, and I. Tudosa. A wi-fi iot prototype for ecg monitoring exploiting a novel compressed sensing method. *ACTA IMEKO*, 9:38. 06 2020.

[14] E. Balestrieri, L. De Vito, F. Picariello, and I. Tudosa. A novel method for compressed sensing based sampling of ecg signals in medical-iot era. In *2019 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 1–6. IEEE, 2019.

[15] S. Baltes and S. Diehl. Towards a theory of software development expertise. In *Proceedings of the 2018 26th acm joint meeting on european software engineering conference and symposium on the foundations of software engineering*, pages 187–200, 2018.

[16] I. Barandiaran. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell*, 20(8):1–22. 1998.

[17] R. G. Baraniuk. Compressive sensing [lecture notes]. *IEEE signal processing magazine*, 24(4):118–121. IEEE, 2007.

[18] P. Bera, R. Gupta, and J. Saha. Preserving Abnormal Beat Morphology in Long-term ECG Recording: An Efficient Hybrid Compression Approach. *IEEE Transactions on Instrumentation and Measurement*, 69(5). IEEE, 2019.

[19] G. Bernardi, L. Cecchetti, G. Handjaras, L. Sani, A. Gaglianese, R. Ceccarelli, F. Franzoni, F. Galetta, G. Santoro, R. Goebel, et al. It's not all in your car: functional and structural correlates of exceptional driving skills in professional racers. *Frontiers in human neuroscience*, 8:888. Frontiers, 2014.

[20] G. Bernardi, L. Cecchetti, G. Handjaras, L. Sani, A. Gaglianese, R. Ceccarelli, F. Franzoni, F. Galetta, G. Santoro, R. Goebel, E. Ricciardi, and P. Pietrini. It's not all in your car: functional and structural correlates of exceptional driving skills in professional racers. *Frontiers in Human Neuroscience*. 2014.

[21] G. Bernardi, E. Ricciardi, L. Sani, A. Gaglianese, A. Papasogli, R. Ceccarelli, F. Franzoni, F. Galetta, G. Santoro, R. Goebel, and P. Pietrini. Functional reorganization of visuo-motor cortical networks in formula 1 pilots versus naive drivers. In *Proceedings of the 17th Annual Meeting of the Organization for Human Brain Mapping*, 2011.

[22] G. Bernardi, E. Ricciardi, L. Sani, A. Gaglianese, A. Papasogli, R. Ceccarelli, F. Franzoni, F. Galetta, G. Santoro, R. Goebel, and P. Pietrini. How skill expertise shapes the brain functional architecture: An fmri study of visuo-spatial and motor processing in professional racing-car and naive drivers. *Plos One journal*. 2013.

[23] L. Billeci, F. Chiarugi, M. Costi, D. Lombardi, and M. Varanini. Detection of af and other rhythms using rr variability and ecg spectral measures. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE, 2017.

[24] R. R. Bouckaert. Bayesian network classifiers in weka. Department of Computer Science, 2004.

[25] M. Bradley. Pj: Measuring emotion: The self-assessment manikin (sam) and the semantic differential. *Journal of Experimental Psychiatry Behavior Therapy*, 25(1):4–59. 1994.

[26] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32. 2001.

[27] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[28] E. Buccelletti, E. Gilardi, E. Scaini, L. Galiuto, R. Persiani, A. Biondi, F. Basile, N. G. Silveri, et al. Heart rate variability and myocardial infarction: systematic literature review and metanalysis. *Eur Rev Med Pharmacol Sci*, 13(4):299–307. Citeseer, 2009.

[29] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27. Taylor & Francis, 1974.

[30] A. J. Camm, G. Corbucci, and L. Padeletti. Usefulness of continuous electrocardiographic monitoring for atrial fibrillation. *The American journal of cardiology*, 110(2):270–276. Elsevier, 2012.

[31] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509. IEEE, 2006.

[32] A. Capucci, G. Calcagnini, E. Mattei, M. Triventi, P. Bartolini, G. Biancalana, A. Gargaro, A. Puglisi, and F. Censi. Daily distribution of atrial arrhythmic episodes in sick sinus syndrome patients: implications for atrial arrhythmia monitoring. *Europace*, 14(8):1117–1124. Oxford University Press, 2012.

[33] F. Censi, G. Calcagnini, E. Mattei, A. Gargaro, G. Biancalana, and A. Capucci. Simulation of monitoring strategies for atrial arrhythmia detection. *Annali dell'Istituto superiore di sanita*, 49:176–182. SciELO Public Health, 2013.

[34] Y. Chang. Reorganization and plastic changes of the human brain associated with skill learning and expertise. *Frontiers in human neuroscience*, 8:35. Frontiers, 2014.

[35] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357. 2002.

[36] S. Chen, W. Hua, Z. Li, J. Li, and X. Gao. Heartbeat classification using projected and dynamic features of ecg signal. *Biomedical Signal Processing and Control*, 31:165–173. Elsevier, 2017.

[37] R. Chitra and V. Seenivasagam. Review of heart disease prediction system using data mining and hybrid intelligent techniques. *ICTACT journal on soft computing*, 3(04):605–09. 2013.

[38] T. Chou, Y. Tamura, and I. Wong. Detection of atrial fibrillation in ecgs. *Computer Methods and Programs in Biomedicine*, 136:143–50. 2016.

[39] S. A. Chouakri, O. Djaafri, and A. Taleb-Ahmed. Wavelet transform and huffman coding based electrocardiogram compression algorithm: Application to telecardiology. *Journal of Physics: Conference Series*, 454:012086. IOP Publishing, aug 2013.

[40] I. Christov, R. Raikova, and S. Angelova. Separation of electrocardiographic from electromyographic signals using dynamic filtration. *Medical engineering & physics*, 57:1–10. Elsevier, 2018.

[41] A. L. Clark, K. Goode, and J. G. Cleland. The prevalence and incidence of left bundle branch block in ambulant patients with chronic heart failure. *European journal of heart failure*, 10(7):696–702. Wiley Online Library, 2008.

[42] G. D. Clifford, F. Azuaje, P. McSharry, et al. *Advanced methods and tools for ECG data analysis*. Artech house Boston, 2006.

[43] W. W. Cohen. Fast effective rule induction. In *Machine learning proceedings 1995*, pages 115–123. Elsevier, 1995.

[44] W. W. Cohen. Fast effective rule induction. In *Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.

[45] J. J. Col and S. L. Weinberg. The incidence and mortality of intraventricular conduction defects in acute myocardial infarction. *The American journal of cardiology*, 29(3):344–350. Elsevier, 1972.

[46] R. Colloca, A. E. Johnson, L. Mainardi, and G. D. Clifford. A support vector machine approach for reliable detection of atrial fibrillation events. In *Computing in Cardiology 2013*, pages 1047–1050. IEEE, 2013.

[47] D. Connaghan, P. Kelly, and N. E. O'Connor. Game, shot and match: Event-based indexing of tennis. In *2011 9th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 97–102. IEEE, 2011.

[48] L. Cordeiro, P. Rabelo, M. Moraes, F. Teixeira-Coelho, C. Coimbra, S. Wanner, and D. Soares. Physical exercise-induced fatigue: the role of serotonergic and dopaminergic systems. *Brazilian journal of medical and biological research*, 50(12). SciELO Brasil, 2017.

[49] R. Couceiro, G. Duarte, J. Durães, J. Castelhano, C. Duarte, C. A. D. Teixeira, M. Castelo-Branco, P. Carvalho, and H. Madeira. Pupillography as indicator of programmers' mental effort and cognitive overload. In *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2019, Portland, OR, USA, June 24-27, 2019*, pages 638–644. IEEE, 2019.

[50] J. S. Cramer. The origins of logistic regression (technical report). In *Tinbergen Institute*, 2002.

[51] P. Cunningham and S. J. Delany. k-nearest neighbour classifiers–. *arXiv preprint arXiv:2004.04523*. 2020.

[52] A. Cutler, D. R. Cutler, and J. R. Stevens. Random forests. In *Ensemble machine learning*, pages 157–175. Springer, 2012.

[53] H. G. B. M. D. B. D. Limmer, M. O'Keefe. *Emergency Care Workbook*. Pearson, 2005.

[54] G. Da Poian, C. J. Rozell, R. Bernardini, R. Rinaldo, and G. D. Clifford. Matched filtering for heart rate estimation on compressive sensing ecg measurements. *IEEE Transactions on Biomedical Engineering*, 65(6):1349–1358. IEEE, 2017.

[55] M. Dai, X. Xiao, X. Chen, H. Lin, W. Wu, and S. Chen. A low-power and miniaturized electrocardiograph data collection system with smart textile electrodes for monitoring of cardiac function. *Australasian physical & engineering sciences in medicine*, 39(4):1029–1040. Springer, 2016.

[56] B. V. Dasarathy. Nearest neighbor (nn) norms: Nn pattern classification techniques. *IEEE Computer Society Tutorial*. 1991.

[57] V. David, N. Adochiei, F. Adochiei, and I. Tudosa. ECG waves and features extraction using Wavelet Multi-Resolution Analysis. In *2011 E-Health and Bioengineering Conference (EHB)*, pages 1–4, 2011.

[58] V. David, N. Adochiei, and I. Tudosa. Methods of electromagnetic interference reduction in electrocardiographic signal acquisition. *Environmental Engineering and Management Journal*, 10(4):553–559. 2011.

[59] P. De Chazal, M. O'Dwyer, and R. B. Reilly. Automatic classification of heartbeats using ecg morphology and heartbeat interval features. *IEEE transactions on biomedical engineering*, 51(7):1196–1206. IEEE, 2004.

[60] C. L. Devasena. Comparative analysis of random forest rep tree and j48 classifiers for credit risk prediction. In *International Conference on Communication, Computing and Information Technology (ICCCMIT-2014)*, 2014.

[61] D. L. Donoho. Compressed sensing: Ieee transactions on information theory. *vol*, 52:1289–1306. 2006.

[62] W. B. Edwards. Modeling overuse injuries in sport as a mechanical fatigue phenomenon. *Exercise and sport sciences reviews*, 46(4):224–231. LWW, 2018.

[63] F. A. Elhaj, N. Salim, A. R. Harris, T. T. Swee, and T. Ahmed. Arrhythmia recognition and classification using combined linear and nonlinear features of ecg signals. *Computer methods and programs in biomedicine*, 127:52–63. Elsevier, 2016.

[64] A. Evans, I. Perez, G. Yu, and L. Kalra. Secondary stroke prevention in atrial fibrillation: lessons from clinical practice. *Stroke*, 31(9):2106–2111. Am Heart Assoc, 2000.

[65] F. Fagerholm and T. Fritz. Biometric measurement in software engineering. In *Contemporary Empirical Methods in Software Engineering*, pages 151–172. Springer, 2020.

[66] G. J. Fahy, S. L. Pinski, D. P. Miller, N. McCabe, C. Pye, M. J. Walsh, and K. Robinson. Natural history of isolated bundle branch block. *The American journal of cardiology*, 77(14):1185–1190. Elsevier, 1996.

[67] S. Fakhoury, Y. Ma, V. Arnaoudova, and O. Adesope. The effect of poor source code lexicon and readability on developers' cognitive load. In *2018 IEEE/ACM 26th International Conference on Program Comprehension (ICPC)*, pages 286–28610. IEEE, 2018.

[68] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Thirteenth International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1996.

[69] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139. Elsevier, 1997.

[70] B. Friedlander and B. Porat. The modified yule-walker method of arma spectral estimation. *IEEE Transactions on Aerospace and Electronic Systems*, (2):158–173. IEEE, 1984.

[71] T. Fritz, A. Begel, S. C. Müller, S. Yigit-Elliott, and M. Züger. Using psycho-physiological measures to assess task difficulty in software development. In *Proceedings of the 36th international conference on software engineering*, pages 402–413, 2014.

[72] T. Fritz, A. Begel, S. C. Müller, S. Yigit-Elliott, and M. Züger. Using Psycho-physiological Measures to Assess Task Difficulty in Software Development. In *Proceedings of the 36th International Conference on Software Engineering*, pages 402–413. ACM, 2014.

[73] T. Fritz and S. C. Müller. Leveraging biometric data to boost software developer productivity. In *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, pages 66–77. IEEE, 2016.

[74] D. Fucci, D. Girardi, N. Novielli, L. Quaranta, and F. Lanubile. A replication study on code comprehension and expertise using lightweight biometric sensors. In *Proceedings of the 27th International Conference on Program Comprehension, ICPC 2019, Montreal, QC, Canada, May 25-31, 2019*, pages 311–322, 2019.

[75] V. Fuster, L. E. Rydén, D. S. Cannom, H. J. Crijns, A. B. Curtis, K. A. Ellenbogen, J. L. Halperin, J.-Y. Le Heuzey, G. N. Kay, J. E. Lowe, et al. Acc/aha/esc 2006 guidelines for the management of patients with atrial fibrillation: A report of the american college of cardiology/american heart association task force on practice guidelines and the european society of cardiology committee for practice guidelines (writing committee to revise the 2001 guidelines for the management of patients with atrial fibrillation): Developed in collaboration with the european heart rhythm association and the heart rhythm society. *Circulation*, 114(7):e257–e354. Am Heart Assoc, 2006.

[76] W. Gao, S. Emaminejad, H. Y. Y. Nyein, S. Challa, K. Chen, A. Peck, H. M. Fahad, H. Ota, H. Shiraki, D. Kiriya, et al. Fully integrated wearable sensor arrays for multiplexed in situ perspiration analysis. *Nature*, 529(7587):509–514. Nature Publishing Group, 2016.

[77] G. Garcia, G. Moreira, D. Menotti, and E. Luz. Inter-patient ecg heartbeat classification with temporal vcg optimized by pso. *Scientific Reports*, 7(1):1–11. Nature Publishing Group, 2017.

[78] A. Ghaemi, M. Rezaie-Balf, J. Adamowski, O. Kisi, and J. Quilty. On the applicability of maximum overlap discrete wavelet transform integrated with mars and m5 model tree for monthly pan evaporation prediction. *Agricultural and Forest Meteorology*, 278:107647. Elsevier, 2019.

[79] D. Girardi, F. Lanubile, and N. Novielli. Emotion detection using noninvasive low cost sensors. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 125–130. IEEE, 2017.

[80] D. Girardi, N. Novielli, D. Fucci, and F. Lanubile. Recognizing developers' emotions while programming. *arXiv preprint arXiv:2001.09177*. 2020.

[81] D. Girardi, N. Novielli, D. Fucci, and F. Lanubile. Recognizing Developers' Emotions while Programming. In *42nd Int. Conf. on Software Engineering (ICSE '20), May 23–29, 2020, Seoul, Republic of Korea*, pages –, 2020.

[82] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220. Am Heart Assoc, 2000.

[83] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220. Am Heart Assoc, 2000.

[84] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220. Am Heart Assoc, 2000.

[85] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson. What happens when software developers are (un) happy. *Journal of Systems and Software*, 140:32–47. Elsevier, 2018.

[86] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson. What happens when software developers are (un)happy. *Journal of Systems and Software*, 140:32 – 47. 2018.

[87] D. Graziotin, X. Wang, and P. Abrahamsson. Software developers, moods, emotions, and performance. *arXiv preprint arXiv:1405.4422*. 2014.

[88] D. Graziotin, X. Wang, and P. Abrahamsson. How do you feel, developer? an explanatory theory of the impact of affects on programming performance. *PeerJ Computer Science*, 1:e18. PeerJ Inc., 2015.

[89] A. Greco, G. Valenza, A. Lanata, E. P. Scilingo, and L. Citi. cvxeda: A convex optimization approach to electrodermal activity processing. *IEEE Transactions on Biomedical Engineering*, 63(4):797–804. IEEE, 2015.

[90] S. Greene, H. Thapliyal, and A. Caban-Holt. A survey of affective computing for stress detection: Evaluating technologies in stress detection for better health. *IEEE Consumer Electronics Magazine*, 5(4):44–56. IEEE, 2016.

[91] M. Gutlein, E. Frank, M. Hall, and A. Karwath. Large-scale attribute selection using wrappers. In *2009 IEEE symposium on computational intelligence and data mining*, pages 332–339. IEEE, 2009.

[92] E. Haapalainen, S. Kim, J. F. Forlizzi, and A. K. Dey. Psycho-physiological measures for assessing cognitive load. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 301–310, 2010.

[93] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18. ACM New York, NY, USA, 2009.

[94] J. Hallberg, S. Svensson, A. Ostmark, P. Lindgren, K. Synnes, and J. Delsing. Enriched media-experience of sport events. In *Sixth IEEE Workshop on Mobile Computing Systems and Applications*, pages 2–9. IEEE, 2004.

[95] A. Haque, M. H. Ali, M. A. Kiber, M. T. Hasan, et al. Detection of small variations of ecg features using wavelet. *ARPN Journal of Engineering and Applied Sciences*, 4(6):27–30. 2009.

[96] A. F. Haque, M. H. Ali, M. A. Kiber, and M. T. Hasan+. Automatic feature extraction of ecg signal using fast fourier transform. June 2009.

[97] R. G. Hart. Atrial fibrillation and stroke prevention. *New England Journal of Medicine*, 349(11):1015–1016. Mass Medical Soc, 2003.

[98] R. G. Hart, J. L. Halperin, L. A. Pearce, D. C. Anderson, R. A. Kronmal, R. McBride, E. Nasco, D. G. Sherman, R. L. Talbert, and J. R. Marler. Lessons from the stroke prevention in atrial fibrillation trials, 2003.

[99] A. E. Hassan. Predicting faults using the complexity of code changes. In *2009 IEEE 31st international conference on software engineering*, pages 78–88. IEEE, 2009.

[100] G. E. Hinton. Connectionist learning procedures. In *Machine learning*, pages 555–610. Elsevier, 1990.

[101] T. K. Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844. IEEE, 1998.

[102] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780. MIT Press, 1997.

[103] C. Huang, S. Ye, H. Chen, D. Li, F. He, and Y. Tu. A novel method for detection of the transition between atrial fibrillation and sinus rhythm. *IEEE Transactions on Biomedical Engineering*, 58(4):1113–1119. IEEE, 2010.

[104] Y. Ikutani and H. Uwano. Brain activity measurement during program comprehension with nirs. In *15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 1–6. IEEE, 2014.

[105] Y. Ikutani and H. Uwano. Brain Activity Measurement During Program Comprehension with NIRS. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2014 15th IEEE/ACIS International Conference on*, pages 1–6. IEEE, 2014.

[106] R. Imanishi, S. Seto, S. Ichimaru, E. Nakashima, K. Yano, and M. Akahoshi. Prognostic significance of incident complete left bundle branch block observed over a 40-year period. *The American journal of cardiology*, 98(5):644–648. Elsevier, 2006.

[107] J. E. Ip and B. B. Lerman. Idiopathic malignant premature ventricular contractions. *Trends in Cardiovascular Medicine*, 28(4):295–302. Elsevier, 2018.

[108] S. T. Iqbal, X. S. Zheng, and B. P. Bailey. Task-evoked pupillary response to mental workload in human-computer interaction. In *CHI'04 extended abstracts on Human factors in computing systems*, pages 1477–1480, 2004.

[109] S. Jaffard, B. Lashermes, and P. Abry. Wavelet leaders in multifractal analysis. In *Wavelet analysis and applications*, pages 201–246. Springer, 2006.

[110] C. T. January, L. S. Wann, J. S. Alpert, H. Calkins, J. E. Cigarroa, J. C. Cleveland, J. B. Conti, P. T. Ellinor, M. D. Ezekowitz, M. E. Field, et al. 2014 aha/acc/hrs guideline for the management of patients with atrial fibrillation: a report of the american college of cardiology/american heart association task force on practice guidelines and the heart rhythm society. *Journal of the American College of Cardiology*, 64(21):e1–e76. Journal of the American College of Cardiology, 2014.

[111] C. K. Jha and M. H. Kolekar. Electrocardiogram data compression using DCT based discrete orthogonal Stockwell transform. *Biomedical Signal Processing and Control*, 46:174–181. Elsevier Ltd, 2018.

[112] T. Jiang, L. Tan, and S. Kim. Personalized defect prediction. In *2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 279–289. Ieee, 2013.

[113] X.-Y. Jing, S. Ying, Z.-W. Zhang, S.-S. Wu, and J. Liu. Dictionary learning based software defect prediction. In *Proceedings of the 36th International Conference on Software Engineering*, pages 414–423, 2014.

[114] D. G. Julian, P. A. Valentine, and G. G. Miller. Disturbances of rate, rhythm and conduction in acute myocardial infarction: a prospective study of 100 consecutive unselected patients with the aid of electrocardiographic monitoring. *The American journal of medicine*, 37(6):915–927. Elsevier, 1964.

[115] Y. Jung and H. Kim. Detection of pvc by using a wavelet-based statistical ecg monitoring procedure. *Biomedical Signal Processing and Control*, 36:176–182. Elsevier, 2017.

[116] R. N. Kandala, R. Dhuli, P. Pławiak, G. R. Naik, H. Moeinzadeh, G. D. Gargiulo, and S. Gunnam. Towards real-time heartbeat classification: evaluation of nonlinear morphological features and voting method. *Sensors*, 19(23):5079. Multidisciplinary Digital Publishing Institute, 2019.

[117] F. Karim, S. Majumdar, H. Darabi, and S. Chen. Lstm fully convolutional networks for time series classification. *IEEE access*, 6:1662–1669. IEEE, 2017.

[118] K. Karkazis and J. R. Fishman. Tracking us professional athletes: The ethics of biometric technologies. *The American Journal of Bioethics*, 17(1):45–60. Taylor & Francis, 2017.

[119] K. Kawamoto, C. A. Houlihan, E. A. Balas, and D. F. Lobach. Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. *Bmj*, 330(7494):765. British Medical Journal Publishing Group, 2005.

[120] K. Kearley, M. Selwood, A. Van den Bruel, M. Thompson, D. Mant, F. R. Hobbs, D. Fitzmaurice, and C. Heneghan. Triage tests for identifying atrial fibrillation in primary care: a diagnostic accuracy study comparing single-lead ecg and modified bp monitors. *BMJ open*, 4(5):e004565. British Medical Journal Publishing Group, 2014.

[121] K. Kevic, B. M. Walters, T. R. Shaffer, B. Sharif, D. C. Shepherd, and T. Fritz. Tracing software developers' eyes and interactions for change tasks. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pages 202–213, 2015.

[122] I. A. Khan, W.-P. Brinkman, and R. M. Hierons. Do moods affect programmers' debug performance? *Cognition, Technology & Work*, 13(4):245–258. Springer, 2011.

[123] H.-G. Kim, E.-J. Cheon, D.-S. Bai, Y. H. Lee, and B.-H. Koo. Stress and heart rate variability: A meta-analysis and review of the literature. *Psychiatry investigation*, 15(3):235. Korean Neuropsychiatric Association, 2018.

[124] P. Kirchhof, S. Benussi, D. Kotecha, A. Ahlsson, D. Atar, B. Casadei, M. Castella, H.-C. Diener, H. Heidbuchel, J. Hendriks, et al. 2016 esc guidelines for the management of atrial fibrillation developed in collaboration with eacts. *European journal of cardio-thoracic surgery*, 50(5):e1–e88. Oxford University Press, 2016.

[125] D. Kleyko, E. Osipov, and U. Wiklund. A comprehensive study of complexity and performance of automatic detection of atrial fibrillation: Classification of long ecg recordings based on the physionet computing in cardiology challenge

2017. *Biomedical Physics & Engineering Express*, 6(2):025010. IOP Publishing, 2020.

[126] R. Kohavi, G. H. John, et al. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324. Elsevier Science, 1997.

[127] R. Kones and J. Phillips. Bundle branch block in acute myocardial infarction. current concepts and indications. *Acta cardiologica*, 35(6):469–478. 1980.

[128] M. Kos and I. Kramberger. A wearable device and system for movement and biometric data acquisition for sports applications. *IEEE Access*, 5:6411–6420. IEEE, 2017.

[129] M. S. Kumar, L. Natrayan, R. Hemanth, K. Annamalai, and E. Karthick. Experimental investigations on mechanical and microstructural properties of al 2 o 3/sic reinforced hybrid metal matrix composite. In *IOP Conf Ser Mater Sci Eng*, page 012123, 2018.

[130] R. C. Kurzweil, L. Gibson, P. Albrecht, and P. Grimshaw. Atrial fibrillation detection, Sept. 29 2009. US Patent 7,596,405.

[131] R. C. Kurzweil, L. Gibson, P. Albrecht, and P. Grimshaw. Atrial fibrillation detection and associated methods, Oct. 4 2016. US Patent 9,456,762.

[132] M. Kuutila, M. Mäntylä, U. Farooq, and M. Claes. Time pressure in software engineering: A systematic literature review. *CoRR*, abs/1901.05771. 2019.

[133] O. Lahdenoja, T. Hurnanen, Z. Iftikhar, S. Nieminen, T. Knuutila, A. Saraste, T. Kiviniemi, T. Vasankari, J. Airaksinen, M. Pänkäälä, et al. Atrial fibrillation detection via accelerometer and gyroscope of a smartphone. *IEEE Journal of Biomedical and Health Informatics*, 22(1):108–118. IEEE, 2017.

[134] D. E. Lake and J. R. Moorman. Accurate estimation of entropy in very short physiological time series: the problem of atrial fibrillation detection in implanted ventricular devices. *American Journal of Physiology-Heart and Circulatory Physiology*, 300(1):H319–H325. American Physiological Society Bethesda, MD, 2010.

[135] S. Lang, F. Bravo-Marquez, C. Beckham, M. Hall, and E. Frank. Wekadeeplearning4j: A deep learning package for weka based on deeplearning4j. *Knowledge-Based Systems*, 178:48–50. Elsevier, 2019.

[136] S. Lang, F. Bravo-Marquez, C. Beckham, M. Hall, and E. Frank. Wekadeeplearning4j: A deep learning package for weka based on deeplearning4j. *Knowledge-Based Systems*, 178:48 – 50. Elsevier, 2019.

[137] B. Lashermes, S. Jaffard, and P. Abry. Wavelet leader based multifractal analysis. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, pages iv–161. IEEE, 2005.

[138] G. Laudato, F. Boldi, A. Colavita, G. Rosa, S. Scalabrino, P. Torchitti, A. Lazich, and R. Oliveto. Combining rhythmic and morphological ecg features for automatic

detection of atrial fibrillation. In *13th International Conference on Health Informatics*, pages 156–165, 2020.

[139] G. Laudato, F. Boldi, A. R. Colavita, G. Rosa, S. Scalabrino, P. Torchitti, A. Lazich, and R. Oliveto. Combining rhythmic and morphological ecg features for automatic detection of atrial fibrillation. In *HEALTHINF*, pages 156–165, 2020.

[140] G. Laudato, G. Rosa, S. Scalabrino, J. Simeone, F. Picariello, I. Tudosa, L. De Vito, F. Boldi, P. Torchitti, R. Ceccarelli, et al. Miphas: Military performances and health analysis system. In *HEALTHINF*, pages 198–207, 2020.

[141] S. le Cessie and J. van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201. 1992.

[142] J.-Y. Le Heuzey, O. Paziaud, O. Piot, M. A. Said, X. Copie, T. Lavergne, and L. Guize. Cost of care distribution in atrial fibrillation patients: the cocaf study. *American heart journal*, 147(1):121–126. Elsevier, 2004.

[143] J. Lee, Y. Nam, D. D. McManus, and K. H. Chon. Time-varying coherence function for atrial fibrillation detection. *IEEE Transactions on Biomedical Engineering*, 60(10):2783–2793. IEEE, 2013.

[144] J. Lee, Y. Nam, D. D. McManus, and K. H. Chon. Time-varying coherence function for atrial fibrillation detection. *IEEE Transactions on Biomedical Engineering*, 60(10):2783–2793. IEEE, 2013.

[145] J. Lee, B. A. Reyes, D. D. McManus, O. Maitas, and K. H. Chon. Atrial fibrillation detection using an iphone 4s. *IEEE Transactions on Biomedical Engineering*, 60(1):203–206. IEEE, 2012.

[146] S. Lee, D. Hooshyar, H. Ji, K. Nam, and H. Lim. Mining biometric data to predict programmer expertise and task difficulty. *Clust. Comput.*, 21(1):1097–1107. 2018.

[147] R. F. Leonarduzzi, G. Schlotthauer, and M. E. Torres. Wavelet leader based multifractal analysis of heart rate variability during myocardial ischaemia. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 110–113. IEEE, 2010.

[148] T. Li and M. Zhou. Ecg classification using wavelet packet entropy and random forests. *Entropy*, 18(8):285. Multidisciplinary Digital Publishing Institute, 2016.

[149] J. Lian, L. Wang, and D. Muessig. A simple method to detect atrial fibrillation using rr intervals. *The American journal of cardiology*, 107(10):1494–1497. Elsevier, 2011.

[150] X. Luo. Ecg signal analysis for fatigue and abnormal event detection during sport and exercise. *Internet Technology Letters*. Wiley Online Library.

[151] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pages 281–297. Oakland, CA, USA, 1967.

[152] M. V. Mäntylä, K. Petersen, T. O. Lehtinen, and C. Lassenius. Time pressure: a controlled experiment of test case development and requirements review. In *Proceedings of the 36th International Conference on Software Engineering*, pages 83–94, 2014.

[153] T. Mar, S. Zaunseder, J. P. Martínez, M. Llamedo, and R. Poll. Optimization of ecg classification by means of feature selection. *IEEE transactions on Biomedical Engineering*, 58(8):2168–2177. IEEE, 2011.

[154] A. Melgarejo-Moreno, J. Galcera-Tomas, A. Garcia-Alberola, M. Valdes-Chavarri, F. J. Castillo-Soria, E. Mira-Sanchez, J. Gil-Sanchez, and J. Allegue-Gallego. Incidence, clinical characteristics, and prognostic significance of right bundle-branch block in acute myocardial infarction: a study in the thrombolytic era. *Circulation*, 96(4):1139–1144. Am Heart Assoc, 1997.

[155] T. Menzies, A. Butcher, D. Cok, A. Marcus, L. Layman, F. Shull, B. Turhan, and T. Zimmermann. Local versus global lessons for defect prediction and effort estimation. *IEEE Transactions on software engineering*, 39(6):822–834. IEEE, 2012.

[156] Miyasaka. Secular trends in incidence of atrial fibrillation in olmsted county, minnesota, 1980 to 2000, and implications on the projections for future prevalence (vol 114, pg 119, 2006). *Circulation*, 114(11):E498–E498. LIPPINCOTT WILLIAMS & WILKINS 530 WALNUT ST, PHILADELPHIA, PA 19106-3621 USA, 2006.

[157] R. Moddemeijer. On estimation of entropy and mutual information of continuous distributions. *Signal processing*, 16(3):233–248. Elsevier, 1989.

[158] M. Mohebbi and H. Ghassemian. Detection of atrial fibrillation episodes using svm. In *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 177–180. IEEE, 2008.

[159] V. Mondéjar-Guerra, J. Novo, J. Rouco, M. G. Penedo, and M. Ortega. Heartbeat classification fusing temporal and morphological information of ecgs via ensemble of classifiers. *Biomedical Signal Processing and Control*, 47:41–48. Elsevier, 2019.

[160] G. B. Moody and R. G. Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50. IEEE, 2001.

[161] D. Morariu, R. Creţulescu, and M. Breazu. The weka multilayer perceptron classifier. *International Journal of Advanced Statistics and IT&C for Economics and Life Sciences*, 7(1). 2018.

[162] J. D. Morris. Observations: Sam: the self-assessment manikin; an efficient cross-cultural measurement of emotional response. *Journal of advertising research*, 35(6):63–68. Advertising Research Foundation, 1995.

[163] J. Muhlsteff, O. Such, R. Schmidt, M. Perkuhn, H. Reiter, J. Lauter, J. Thijs, G. Musch, and M. Harris. Wearable approach for continuous ecg-and activity

patient-monitoring. In *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2184–2187. IEEE, 2004.

[164] S. Mukhopadhyay, S. Mitra, and M. Mitra. An ecg signal compression technique using ascii character encoding. *Measurement*, 45(6):1651 – 1660. 2012.

[165] S. Müller and T. Fritz. Stuck and Frustrated or in Flow and Happy: Sensing Developers' Emotions and Progress. In *Software Engineering (ICSE), 2015 IEEE/ACM 37th IEEE International Conference on*, pages 688–699. IEEE, 2015.

[166] S. Müller and T. Fritz. Using (bio) Metrics to Predict Code Quality Online. In *Proceedings of the 38th International Conference on Software Engineering*, pages 452–463. ACM, 2016.

[167] S. C. Müller and T. Fritz. Stuck and frustrated or in flow and happy: sensing developers' emotions and progress. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, pages 688–699. IEEE, 2015.

[168] S. C. Müller and T. Fritz. Using (bio) metrics to predict code quality online. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*, pages 452–463. IEEE, 2016.

[169] C. B. Mullins and J. M. Atkins. Prognoses and management of venticular conduction blocks in acute myocardial infarction. *Modern Concepts of Cardiovascular Disease*, 45(10):129–133. 1976.

[170] N. Nagappan, T. Ball, and A. Zeller. Mining metrics to predict component failures. In *Proceedings of the 28th international conference on Software engineering*, pages 452–461, 2006.

[171] T. Nakagawa, Y. Kamei, H. Uwano, A. Monden, K. Matsumoto, and D. M. German. Quantifying programmers' mental workload during program comprehension based on cerebral blood flow measurement: a controlled experiment. In *Companion proceedings of the 36th international conference on software engineering*, pages 448–451, 2014.

[172] L. Natrayan, M. S. Kumar, and K. Palanikumar. Optimization of squeeze cast process parameters on mechanical properties of al2o3/sic reinforced hybrid metal matrix composites using taguchi technique. *Materials Research Express*, 5(6):066516. IOP Publishing, 2018.

[173] K. H. Newby, E. Pisano, M. W. Krucoff, C. Green, and A. Natale. Incidence and clinical relevance of the occurrence of bundle-branch block in patients treated with thrombolytic therapy. *Circulation*, 94(10):2424–2428. Am Heart Assoc, 1996.

[174] W. S. Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567. Nature Publishing Group, 2006.

[175] N. Owens, C. Harris, and C. Stennett. Hawk-eye tennis system. In *2003 International Conference on Visual Information Engineering VIE 2003*, pages 182–185. IET, 2003.

[176] S. K. Pal and S. Mitra. Multilayer perceptron, fuzzy sets, classifiaction. 1992.

[177] S. K. Pal and S. Mitra. Multilayer perceptron, fuzzy sets, classifiaction. 1992.

[178] J. Pan and W. J. Tompkins. A real-time qrs detection algorithm. *IEEE Trans. Biomed. Eng*, 32(3):230–236. 1985.

[179] J. Pan and W. J. Tompkins. A real-time qrs detection algorithm. *IEEE transactions on biomedical engineering*, (3):230–236. IEEE, 1985.

[180] J. Pan and W. J. Tompkins. A real-time qrs detection algorithm. *IEEE transactions on biomedical engineering*, (3):230–236. IEEE, 1985.

[181] S. K. Pandey and R. R. Janghel. Automatic arrhythmia recognition from electrocardiogram signals using different feature methods with long short-term memory network model. *Signal, Image and Video Processing*, pages 1–9. Springer, 2020.

[182] C. Parnin. Subvocalization-Toward Hearing the Inner Thoughts of Developers. In *Program Comprehension (ICPC), 2011 IEEE 19th International Conference on*, pages 197–200. IEEE, 2011.

[183] N. Peitek, J. Siegmund, S. Apel, C. Kästner, C. Parnin, A. Bethmann, T. Leich, G. Saake, and A. Brechmann. A Look into Programmers' Heads. *IEEE Transactions on Software Engineering*. IEEE, 2018.

[184] A. Petrėnas, V. Marozas, and L. Sörnmo. Low-complexity detection of atrial fibrillation in continuous long-term monitoring. *Computers in biology and medicine*, 65:184–191. Elsevier, 2015.

[185] A. Petrėnas, V. Marozas, and L. Sörnmo. Low-complexity detection of atrial fibrillation in continuous long-term monitoring. *Computers in biology and medicine*, 65:184–191. Elsevier, 2015.

[186] F. Picariello, G. Iadarola, E. Balestrieri, I. Tudosa, and L. De Vito. A novel compressive sampling method for ecg wearable measurement systems. *Measurement*, 167:108259. 2021.

[187] J. Pilzer, R. Rosenast, A. N. Meyer, E. M. Huang, and T. Fritz. Supporting software developers' focused work on window-based desktops. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

[188] K. C. Pohlmann. *Principles of digital audio*. McGraw-Hill, Inc., 1995.

[189] H. F. Posada-Quintero, J. P. Florian, A. D. Orjuela-Cañón, and K. H. Chon. Electrodermal activity is sensitive to cognitive stress under water. *Frontiers in physiology*, 8:1128. Frontiers, 2018.

[190] J. R. Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[191] S. Radevski, H. Hata, and K. Matsumoto. Real-time Monitoring of Neural State in Assessing and Improving Software Developers' Productivity. In *Proceedings of the Eighth International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 93–96. IEEE Press, 2015.

[192] S. Raj and K. C. Ray. Sparse representation of ecg signals for automated recognition of cardiac arrhythmias. *Expert systems with applications*, 105:49–64. Elsevier, 2018.

[193] R. Rajesh, K.N.and Dhuli. Classification of imbalanced ecg beats using resampling techniques and adaboost ensemble classifier. *Biomed. Signal Process. Control*, 41:242–254. 2018.

[194] K. Rajeswari and R. Anantharaman. Development of an instrument to measure stress among software professionals: Factor analytic study. In *Proceedings of the 2003 SIGMIS conference on Computer personnel research: Freedom in Philadelphia–leveraging differences and diversity in the IT workforce*, pages 34–43, 2003.

[195] L. Rebollo-Neira. Effective high compression of ecg signals at low level distortion. *Scientific reports*, 9(1):1–12. Nature Publishing Group, 2019.

[196] P. Rizzon, M. Di Biase, and C. Baissus. Intraventricular conduction defects in acute myocardial infarction. *British Heart Journal*, 36(7):660. BMJ Publishing Group, 1974.

[197] C. Roopa and B. Harish. A survey on various machine learning approaches for ecg analysis. *International Journal of Computer Applications*, 163(9):25–33. Foundation of Computer Science, 2017.

[198] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65. Elsevier, 1987.

[199] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*, volume 589. John wiley & sons, 2005.

[200] J. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39:1161–1178. 1980.

[201] S. Safdar, S. Zafar, N. Zafar, and N. F. Khan. Machine learning based decision support systems (dss) for heart disease diagnosis: a review. *Artificial Intelligence Review*, 50(4):597–623. Springer, 2018.

[202] S. Schmidt and H. Walach. Electrodermal activity (eda)–state-of-the-art measurements and techniques for parapsychological purposes. *Journal of Parapsychology*, 64(2). 2000.

[203] R. B. Schnabel, X. Yin, P. Gona, M. G. Larson, A. S. Beiser, D. D. McManus, C. Newton-Cheh, S. A. Lubitz, J. W. Magnani, P. T. Ellinor, et al. 50 year

trends in atrial fibrillation prevalence, incidence, risk factors, and mortality in the framingham heart study: a cohort study. *The Lancet*, 386(9989):154–162. Elsevier, 2015.

[204] C. Schubert, M. Lambertz, R. Nelesen, W. Bardwell, J.-B. Choi, and J. Dimsdale. Effects of stress on heart rate complexity—a comparison between short-term and chronic stress. *Biological psychology*, 80(3):325–332. Elsevier, 2009.

[205] H. Sedghamiz. Matlab implementation of pan tompkins ecg qrs detector. *Code Available at the File Exchange Site of MathWorks*. 2014.

[206] H. Sedghamiz. Matlab implementation of pan tompkins ecg qrs detector. *Code Available at the File Exchange Site of MathWorks*. 2014.

[207] J. Seo, Y.-T. Kim, H.-J. Song, H. J. Lee, J. Lee, T.-D. Jung, G. Lee, E. Kwon, J. G. Kim, and Y. Chang. Stronger activation and deactivation in archery experts for differential cognitive strategy in visuospatial working memory processing. *Behavioural brain research*, 229(1):185–193. Elsevier, 2012.

[208] J. Sepulveda-Suescun, J. Murillo-Escobar, R. Urda-Benitez, D. Orrego-Metaute, and A. Orozco-Duque. Atrial fibrillation detection through heart rate variability using a machine learning approach and poincare plot features. In *VII Latin American Congress on Biomedical Engineering CLAIB 2016, Bucaramanga, Santander, Colombia, October 26th-28th, 2016*, pages 565–568. Springer, 2017.

[209] J. Sepulveda-Suescun, J. Murillo-Escobar, R. Urda-Benitez, D. Orrego-Metaute, and A. Orozco-Duque. Atrial fibrillation detection through heart rate variability using a machine learning approach and poincare plot features. In *VII Latin American Congress on Biomedical Engineering CLAIB 2016, Bucaramanga, Santander, Colombia, October 26th-28th, 2016*, pages 565–568. Springer, 2017.

[210] C. Setz, B. Arnrich, J. Schumm, R. La Marca, G. Tröster, and U. Ehlert. Discriminating stress from cognitive load using a wearable eda device. *IEEE Transactions on information technology in biomedicine*, 14(2):410–417. IEEE, 2009.

[211] B. Sharif and T. Shaffer. The use of eye tracking in software development. In *International Conference on Augmented Cognition*, pages 807–816. Springer, 2015.

[212] T. Shaw. The emotions of systems developers: an empirical study of affective events theory. In *Proceedings of the 2004 SIGMIS conference on Computer personnel research: Careers, culture, and ethics in a networked environment*, pages 124–126, 2004.

[213] H. J. Shenkman, V. Pampati, A. K. Khandelwal, J. McKinnon, D. Nori, S. Kaatz, K. R. Sandberg, and P. A. McCullough. Congestive heart failure and qrs duration: establishing prognosis study. *Chest*, 122(2):528–534. Elsevier, 2002.

[214] J. Siegmund, C. Kästner, S. Apel, C. Parnin, A. Bethmann, T. Leich, G. Saake, and A. Brechmann. Understanding understanding source code with functional

magnetic resonance imaging. In *Proceedings of the 36th International Conference on Software Engineering*, pages 378–389, 2014.

[215] J. Siegmund, C. Kästner, S. Apel, C. Parnin, A. Bethmann, T. Leich, G. Saake, and A. Brechmann. Understanding Understanding Source Code with Functional Magnetic Resonance Imaging. In *Proceedings of the 36th International Conference on Software Engineering*, pages 378–389. ACM, 2014.

[216] R. Smisek, I. Viscor, P. Jurak, J. Halamek, and F. Plesinger. Fully automatic detection of strict left bundle branch block. *Journal of Electrocardiology*, 51(6):S31–S34. Elsevier, 2018.

[217] M. Soto, C. Satterfield, T. Fritz, G. C. Murphy, D. C. Shepherd, and N. Kraft. Observing and predicting knowledge worker stress, focus and awakeness in the wild. *International Journal of Human-Computer Studies*, 146:102560. Elsevier.

[218] S. Stewart, N. Murphy, A. Walker, A. McGuire, and J. McMurray. Cost of an emerging epidemic: an economic analysis of atrial fibrillation in the uk. *Heart*, 90(3):286–292. BMJ Publishing Group Ltd, 2004.

[219] S. Stewart, N. Murphy, A. Walker, A. McGuire, and J. McMurray. Cost of an emerging epidemic: an economic analysis of atrial fibrillation in the uk. *Heart*, 90(3):286–292. BMJ Publishing Group Ltd and British Cardiovascular Society, 2004.

[220] J. R. Stroop. Studies of interference in serial verbal reactions. *Journal of Experimental Psychology*, 18(6). 1935.

[221] J. R. Stroop. Studies of interference in serial verbal reactions. *Journal of experimental psychology*, 18(6):643. Psychological Review Company, 1935.

[222] W. M. Suess, A. B. Alexander, D. D. Smith, H. W. Sweeney, and R. J. Marion. The effects of psychological stress on respiration: a preliminary study of anxiety and hyperventilation. *Psychophysiology*, 17(6):535–540. Wiley Online Library, 1980.

[223] M. L. Talbi and P. Ravier. Detection of pvc in ecg signals using fractional linear prediction. *Biomedical Signal Processing and Control*, 23:42–51. Elsevier, 2016.

[224] H. Talukder, T. Vincent, G. Foster, C. Hu, J. Huerta, A. Kumar, M. Malazarte, D. Saldana, and S. Simpson. Preventing in-game injuries for nba players. In *Proceedings of the MIT Sloan Sports Analytics Conference, Boston, MA, USA*, pages 11–12, 2016.

[225] J. J. Tecce. Psychophysiology: Human behavior and physiological response, jl andreassi, lawrence erlbaum associates, mahwah, nj (2007), 2007.

[226] M. Tlili, M. Ben-Romdhane, A. Maalej, F. Rivet, D. Dallet, and C. Rebai. Level-crossing adc design and evaluation methodology for normal and pathological electrocardiogram signals measurement. *Measurement*, 124:413 – 425. 2018.

[227] I. Tomek. Two modifications of cnn. 1976.

[228] G. Trends. Public health and aging: trends in aging—united states and worldwide. *Public Health*, 347:921–5. 2003.

[229] Y. M. Ulrich-Lai and J. P. Herman. Neural regulation of endocrine and autonomic stress responses. *Nature reviews neuroscience*, 10(6):397–409. Nature Publishing Group, 2009.

[230] C. van Walraven, R. G. Hart, D. E. Singer, P. J. Koudstaal, and S. Connolly. Oral anticoagulants vs. aspirin for stroke prevention in patients with non-valvular atrial fibrillation: the verdict is in. *Cardiac electrophysiology review*, 7(4):374–378. Springer, 2003.

[231] R. Villar, T. Beltrame, and R. L. Hughson. Validation of the hexoskin wearable vest during lying, sitting, standing, and walking activities. *Applied Physiology, Nutrition, and Metabolism*, 40(10):1019–1024. NRC Research Press, 2015.

[232] C. H. Vinkers, R. Penning, J. Hellhammer, J. C. Verster, J. H. Klaessens, B. Olivier, and C. J. Kalkman. The effect of stress on core and peripheral body temperature in humans. *Stress*, 16(5):520–530. Taylor & Francis, 2013.

[233] W. von Rosenberg, T. Chanwimalueang, T. Adjei, U. Jaffer, V. Goverdovsky, and D. P. Mandic. Resolving ambiguities in the lf/hf ratio: Lf-hf scatter plots for the categorization of mental and physical stress from hrv. *Frontiers in physiology*, 8:360. Frontiers, 2017.

[234] D. Wallmann, D. Tüller, N. Kucher, J. Fuhrer, M. Arnold, and E. Delacretaz. Frequent atrial premature contractions as a surrogate marker for paroxysmal atrial fibrillation in patients with acute ischaemic stroke. *Heart*, 89(10):1247–1248. BMJ Publishing Group Ltd, 2003.

[235] D. Wallmann, D. Tuller, K. Wustmann, P. Meier, J. Isenegger, M. Arnold, H. P. Mattle, and E. Delacrétaz. Frequent atrial premature beats predict paroxysmal atrial fibrillation in stroke patients: an opportunity for a new diagnostic strategy. *Stroke*, 38(8):2292–2294. Am Heart Assoc, 2007.

[236] D. Wastell and M. Newman. The behavioral dynamics of information system development: A stress perspective. *Accounting, Management and Information Technologies*, 3(2):121–148. Elsevier, 1993.

[237] G. Wei, Y. Zhang, T. Jiang, and J. Luo. Increased cortical thickness in sports experts: a comparison of diving players with the controls. *PLoS One*, 6(2):e17112. Public Library of Science, 2011.

[238] M. R. Wrobel. Emotions in the software development process. In *2013 6th International Conference on Human System Interactions (HSI)*, pages 518–523. IEEE, 2013.

[239] Z. Xiong, M. K. Stiles, and J. Zhao. Robust ecg signal classification for detection of atrial fibrillation using a novel neural network. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE, 2017.

[240] S. S. Xu, M.-W. Mak, and C.-C. Cheung. Towards end-to-end ecg classification with raw signal extraction and deep neural networks. *IEEE journal of biomedical and health informatics*, 23(4):1574–1584. IEEE, 2018.

[241] S. S. Xu, M.-W. Mak, and C.-C. Cheung. Towards end-to-end ecg classification with raw signal extraction and deep neural networks. *IEEE journal of biomedical and health informatics*. IEEE, 2018.

[242] F. Yan, J. Kittler, D. Windridge, W. Christmas, K. Mikolajczyk, S. Cox, and Q. Huang. Automatic annotation of tennis games: An integration of audio, vision, and learning. *Image and Vision Computing*, 32(11):896–903. Elsevier, 2014.

[243] C. Ye, B. V. Kumar, and M. T. Coimbra. Heartbeat classification using morphological and dynamic features of ecg signals. *IEEE Transactions on Biomedical Engineering*, 59(10):2930–2941. IEEE, 2012.

[244] S. Yogeshwaran, R. Prabhu, L. Natrayan, and R. Murugan. Mechanical properties of leaf ashes reinforced aluminum alloy metal matrix composites. *Int. J. Appl. Eng. Res*, 10(13):11048–11052. 2015.

[245] C. Yuan, Y. Yan, L. Zhou, J. Bai, and L. Wang. Automated atrial fibrillation detection based on deep learning network. In *2016 IEEE International Conference on Information and Automation (ICIA)*, pages 1159–1164. IEEE, 2016.

[246] Q. Zhao and L. Zhang. Ecg feature extraction and classification using wavelet transform and support vector machines. In *2005 International Conference on Neural Networks and Brain*, pages 1089–1092. IEEE, 2005.

[247] Q. Zhao and L. Zhang. Ecg feature extraction and classification using wavelet transform and support vector machines. In *2005 International Conference on Neural Networks and Brain*, pages 1089–1092. IEEE, 2005.

[248] X. Zhou, H. Ding, B. Ung, E. Pickwell-MacPherson, and Y. Zhang. Automatic online detection of atrial fibrillation based on symbolic dynamics and shannon entropy. *Biomedical engineering online*, 13(1):18. BioMed Central, 2014.

[249] X. Zhou, H. Ding, W. Wu, and Y. Zhang. A real-time atrial fibrillation detection algorithm based on the instantaneous state of heart rate. *PloS one*, 10(9):e0136544. Public Library of Science, 2015.

[250] N. Zhu, T. Diethe, M. Camplani, L. Tao, A. Burrows, N. Twomey, D. Kaleshi, M. Mirmehdi, P. Flach, and I. Craddock. Bridging e-health and the internet of things: The sphere project. *IEEE Intelligent Systems*, 30(4):39–46. IEEE, 2015.

[251] Y. Zigel, A. Cohen, and A. Katz. Ecg signal compression using analysis by synthesis coding. *IEEE Transactions on Biomedical Engineering*, 47(10):1308–1316. 2000.

[252] M. Zoni-Berisso, F. Lercari, T. Carazza, and S. Domenicucci. Epidemiology of atrial fibrillation: European perspective. *Clinical epidemiology*, 6:213. Dove Press, 2014.

[253] M. Züger and T. Fritz. Interruptibility of software developers and its prediction using psycho-physiological sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2981–2990, 2015.

[254] M. Züger, S. C. Müller, A. N. Meyer, and T. Fritz. Sensing interruptibility in the office: A field study on the use of biometric and computer interaction sensors. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2018.

[255] V. Zurro, A. Stelle, and J. Nadal. Detection of atrial persistent rhythm based on p-wave recognition and rr interval variability. In *Computers in Cardiology 1995*, pages 185–188. IEEE, 1995.