PhD Thesis

Algorithms for Sleep Medicine

Isaac Fernández Varela

2019

UNIVERSIDADE DA CORUÑA

Algorithms for Sleep Medicine

Isaac Fernández Varela

PhD Thesis

June 2019

PhD Advisors:
Elena Hernández Pereira
Vicente Moret Bonillo

PhD in Computational Science

UNIVERSIDADE DA CORUÑA

Dr. Vicente Moret Bonillo
Catedrático de Universidad
Dpto. de Computación
Universidade da Coruña

Dra. Elena Hernández Pereira
Profesora titular de Universidad
Dpto. de Computación
Universidade da Coruña

CERTIFICAN

Que la memoria titulada "*Algorithms for Sleep Medicine*" ha sido realizada por D. Isaac Fernández Varela bajo nuestra dirección en el Departamento de Computación de la Universidade da Coruña, y concluye la Tesis Doctoral que presenta para optar al grado de Doctor en Ingeniería Informática con la Mención de Doctor Internacional.

En A Coruña, a 1 de junio de 2019

MORET BONILLO VICENTE - 32746092A

Firmado digitalmente por MORET BONILLO VICENTE - 32746092A
Nombre de reconocimiento (DN): c=ES, serialNumber=IDCES-32746092A, givenName=VICENTE, sn=MORET BONILLO, cn=MORET BONILLO VICENTE - 32746092A
Fecha: 2019.07.23 11:04:54 +02'00'

Fdo.: Vicente Moret Bonillo
Director de la Tesis Doctoral

Elena Hernández Pereira

Firmado digitalmente por Elena Hernández Pereira
Nombre de reconocimiento (DN): cn=Elena Hernández Pereira, o=Universidade da Coruña, ou=Universidade da Coruña, email=elena.hernandez@udc.es, c=ES
Fecha: 2019.07.24 12:54:55 +02'00'

Fdo.: Elena Hernández Pereira
Directora de la Tesis Doctoral

Fdo.: Isaac Fernández Varela
Autor de la Tesis Doctoral

FERNANDEZ VARELA ISAAC-DANIEL - 45871112G
2019.07.23 10:34:21 +02'00'

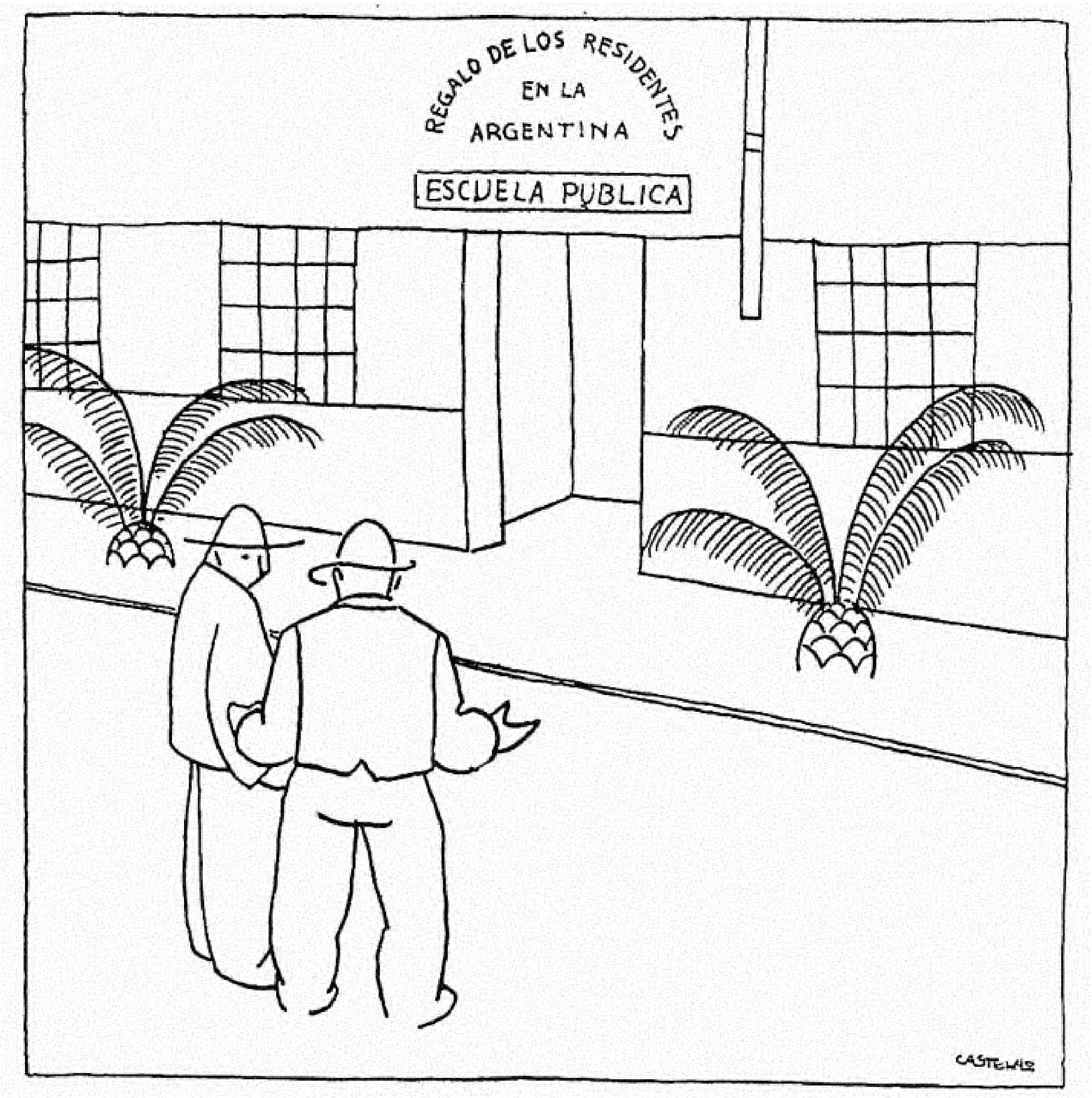
Acknowledgments

I have to start thanking my two advisors. They made this thesis possible. Thank you, Professor Vicente Moret-Bonillo, for giving me the opportunity and letting me be part of your group. Thank you, Dr Elena Hernández-Pereira, for your daily kindness, your faithful support, and for having always helped me going.

Undoubtedly, this adventure would have been much harder without Lidia. I have to name Borja and Laura, as they are special to me, but they all are an amazing group that made working a joy. Long runs are always easier with a partner, and they were the best I could have dreamed of.

I gratefully thank Dr David Martínez Rego and Professor John Shawe-Taylor for hosting me during my visit to the Computer Science Laboratory at the University College of London. In the same way, I gratefully thank Dr R. Rijsman and Dr Diego Alvarez-Estevez for giving me the opportunity to research at the Sleep Center in the Haaglanden Medisch Centrum. I cannot describe my gratitude to you, Diego. You always asked the toughest questions and provided the clearest explanations, making me realise why I love to do research. Thanks for helping me to reach this level and for shaping my mind into always asking the next question.

Finally, I have to thank all the people that trusted in me, those who blindly believe I would get to this point and never ceased to support me. To my friends, my family, my parents, Juana and Ramón, and to her, Diana, thank you. Always.



A escola é boa, ¡Dios o pague! ¿Pero o maiestro é o mesmo que xa tiñamos?

Alfonso Rodríguez Castelao

Abstract

Sleep disorders affect a significant part of our population causing problems that go from daytime sleepiness to severe, life-threatening conditions. Fortunately, physicians can diagnose them and propose a treatment after analyzing the data recorded with a sleep study. The most common one is polysomnography. Neurophysiological signals are recorded during sleep and later analyzed by experts. The goal is the characterization of sleep macro and microstructure to compare it against regular and abnormal sleep characterization, leading to the identification of several sleep disorders. The problem is that this task is complex and tedious because it involves many data. The analysis of a single patient's night data can take several hours even for an expert. Undoubtedly, this time limits the capacity of sleep centers, being the *de facto* bottleneck of these medical units. This thesis addresses this problem. The purpose is to develop algorithms that analyze the signals automatically, discharging the responsibility from the expert. Thus, the expert would only expend time in the diagnosis and development of treatment plans.

We propose methods for the classification of sleep stages and the detection of sleep events. We also present the validation of one of our algorithms and the construction of an API, intended to facilitate the use of our methods.

In this thesis, we use artificial intelligence to meet our goals. With feature extraction and machine learning, we detect two sleep micro-events: arousals and sleep spindles. For the former, we also propose a method using pattern matching. To classify the sleep stages, we mainly rely on deep learning methods.

Resumo

Os trastornos do sono afectan a unha parte importante da sociedade, causando problemas que van dende a somnolencia diúrna ata condicións severas que ameazan a supervivencia. Afortunadamente, os expertos médicos poden diagnosticalos e propoñer un tratamento despois de analizar os datos rexistrados nun estudo do sono. O máis común destes estudos é a polisomnografía. Durante o sono, rexístranse os sinais neurofisiolóxicas do doente e, posteriormente, os expertos estúdanos e analízanos. O obxectivo é caracterizar a macro e micro estrutura do sono para comparala con caracterizacións de referencia, tanto de sono normal como de sono con trastornos. Mediante esta comparación pódese identificar a patoloxía do doente. O problema desta aproximación é que a caracterización é unha tarefa complexa e árida, consumindo varias horas incluso a un experto adestrado. Sen dúbida, a duración de esta tarefa diminúe a capacidade das unidades do sono, sendo o seu límite principal. Nesta tese desenvolvemos algoritmos que analizan os sinais automaticamente, solucionando este problema. Evitamos que o tempo do experto se consuma no análise para que o poida empregar na diagnose e na proposta de tratamento.

Propoñemos métodos para a clasificación das fases do sono e a detección dos seus eventos, cubrindo así a caracterización da macro e micro estrutura do sono. Tamén presentamos a validación dun dos nosos algoritmos, utilizándoo nun entorno real, e a construción dunha API, pensada para facilitar o uso dos nosos algoritmos.

Nesta tese utilizamos a intelixencia artificial para conseguir as nosas metas. Con extracción de características e aprendizaxe máquina detectamos os eventos da microestrutura do sono: despertares e fusos do sono. Para o primeiro evento tamén incluímos un método baseado no recoñecemento de patróns. Para a clasificación das fases do sono utilizamos modelos de aprendizaxe profunda, en particular redes convolucionais.

Resumen

Los trastornos del sueño afectan a una parte significativa de la población, causando problemas que van desde la somnolencia diurna a condiciones severas que amenazan la supervivencia. Afortunadamente, los expertos médicos pueden diagnosticarlos y proponer un tratamiento después de analizar los datos registrados con un estudio del sueño. El más común de estos estudios es la polisomnografía. Durante el sueño, se registran las señales neurofisiológicas del paciente y, posteriormente, las estudian y analizan los expertos. El objetivo es caracterizar la macro estructura y la microestructura del sueño. Comparando la caracterización con las de un sueño normal y los afectados por trastornos, se puede identificar la patología del paciente. El problema de esta aproximación es que la caracterización es una tarea compleja y tediosa, con una duración de horas incluso para el experto entrenado. Indudablemente, la duración de esta tarea limita la capacidad de las unidades de sueño, siendo el principal cuello de botella. En esta tesis desarrollamos algoritmos que analizan las señales automáticamente, solucionando este problema. Evitamos que el tiempo del experto se consuma en el análisis para que lo pueda enfocar en la diagnosis y en la propuesta de tratamiento.

Proponemos métodos para la clasificación de las fases de sueño y la detección de eventos de sueño, cubriendo así la caracterización de la macro y microestructura del sueño. También presentamos la validación de uno de nuestros algoritmos, que se utilizó en un entorno real, y la construcción de una API, pensada para facilitar el uso de nuestros algoritmos.

En esta tesis utilizamos inteligencia artificial para conseguir nuestras metas. Con extracción de características y aprendizaje máquina detectamos dos eventos de la microestructura del sueño: despertares y husos de sueño. Para el primero también incluimos un método basado en el reconocimiento de patrones. Para la clasificación

de las fases del sueño utilizamos modelos de aprendizaje profundo, en concreto redes convolucionales.

Contents

Chapter 1

Introduction

Sleep is the resting state in which the body is not active, and the mind is unconscious. Usually, the body is in a lying posture, there are not voluntary corporal movements, and the response to external stimuli is low. Also, sleep duration has to be limited to some hours, typically between 6 and 10 in humans. Otherwise, we would be talking about other states as comma or hibernation.

Sleep triggers complex mechanisms, some of them still under study, as changes of the hormonal levels, metabolic and biochemical processes and thermoregulation. Although we are not sure about the function of these mechanisms, we know that sleep is fundamental to life. Thus, sleep is still under study from two different approaches. Firstly, considering the physiology of sleep, measuring it and relating the measures with various functions. Secondly, taking behavioral consequences of sleep and attempting to find the physiological measures to explain them. An example of the former was the discovery of slow waves and the attempt to relate them with memory. An example of the latter is how the study of the role of sleep in alertness led to the knowledge of how the hypothalamus is involved in this function.

In any case, it seems that sleep has more than one purpose as memory formation, boost alertness and attention, stabilize mood, reduce strain on joints and muscles, enhance the immune system or signal changes in hormone release. Some of these purposes are altered in the presence of a sleep disorder, causing problems that go from day time sleepiness to life-threatening conditions. Unfortunately, sleep disorders affect a significant part of the population. Just as an example, between 30% and 40% of adults complain of insomnia, and between 5% and 15% of sleepiness [1].

Advances in the knowledge about sleep and sleep disorders facilitate the recognition of Sleep Medicine as a specialty from the second half of the 20th century. Still, nowadays there is no standard on how to train these specialists or how to set up their laboratories [2, 3]. Nevertheless, it is clear that sleep medicine is devoted to the diagnosis and therapy of sleep disorders.

Doctors can diagnose these disorders analyzing data recorded during a sleep study carried out in a sleep laboratory, being the polysomnography (PSG) the most common one. These data are studied and characterized, usually trying to determine the sleep macrostructure, i.e., sleep stages, and the sleep microstructure, i.e., events happening during sleep such as arousals or sleep spindles. With the results of this analysis, the specialist can make a diagnosis and propose a treatment.

The classification of sleep stages requires handling and analyzing large amounts of information and knowledge [4]. Also, the quality and inter-rater agreement is often less than 90%. For example, Stepnowsky et al. [5] studied the agreement between two human raters and found kappas of 0.46–0.89. Similarly, Wang et al. [6] found kappas between 0.72 and 0.85. Furthermore, the agreement between experts is even lower when considering events from sleep microstructure. For example, reported agreement for the detection of arousals is in the range 0.47-0.57 [7, 8].

In this thesis, we describe algorithms that automatically characterized the data obtained in PSG studies. Why? Firstly, because this is typically the most consuming task, so it is usually the bottleneck limiting the capacity of sleep laboratories. Secondly, because these algorithms would improve cohesion, improving the agreement between different sleep centers.

This introduction describes our domain, beginning with a brief history of sleep medicine. Afterward, we introduce sleep studies, focusing on polysomnography. Then we describe the sleep macro and microstructure, with an emphasis in sleep stages, arousals, and sleep spindles. We also summarise the research presented in the subsequent chapters. We finish this chapter with conclusions and future work.

1.1 A brief history of sleep medicine

Although documents that refer to sleep date as back as to ancient Egypt [9], it was not until 1913 that Henri Piéron [10] published the first book that attempted to deal with the physiology of sleep. Two years before, he had discovered a molecule that could induce sleep when injected into animals [11]. In 1916, Constantin von Economo identified the hypothalamus, considering it the center of sleep and wake activity [12]. In 1925, one of the fathers of sleep medicine, Nathaniel Kleitman, published his first work. Although his most significant contribution, the discovery of the rapid eye movement stage (REM), would not happen until 1953 [13]. The next big leap occurs in 1937, when Loomis, Harvey, and Hobart identified five sleep stages using the electroencephalogram (invented in 1924), naming brain waves and defining their main characteristics [14]. Their work was, indeed, the seed of nowadays sleep structure. Another important book, *Sleep and Wakefulness* by Kleitman, was published in 1939 [15]. The book covered sleep research, sleep disorders, changes in body temperature and sleep-wake cycles. Kleitman also discovered, in 1954, the fact that sleep itself is also a repetition of cycles. Later, in 1959, Michel Jouvet made the distinction between REM and non-REM sleep, taking into account the variation of the brain and muscle activity [16].

Around the sixties, obstructive sleep apnea was the first sleep disorder studied in detail, also describing the physiological changes that it implied. This research was afterward extended to a systematic study of temperature, circulatory, and breathing changes during sleep. Later, parasomnias and bed-wetting were associated with awakening from slow-wave sleep. Following this detailed study of disorders, Stanford opened the first sleep research center in 1970. Two years before, Rechtschaffen and Kales had published *A Manual of Standardised Terminology, Techniques and Scoring System for Sleep Stages of Human Subjects* [17], the classification of sleep stages that was in use until 2007.

The confirmation of the necessity of sleep to life would wait until 1983 when researchers showed that sleep-deprived rats suffered severe health consequences [18]. In 1988, researchers carried out the first sleep cohort study [19], finding that the prevalence of sleep apnea in the American population was between 2 and 4%. Soon after, in 1989, the first book on sleep medicine, *Principles of Sleep Medicine* [20], was published.

The first general diagnostic tool for sleep disorders was presented in 1991: the Epworth sleepiness scale, a subjective assessment of daytime sleepiness [21]. In the following years, the knowledge of sleep kept growing, showing its relationship with energy and metabolism. It was in 1999 when the American Academy of Sleep

Medicine (AASM) appeared. Even then, the association already included over 7000 physicians, researchers, and professionals specialized in sleep medicine. In 2007, they published the latest classification of sleep stages, which with minor changes is the standard today [22].

From there, researchers discovered new relationships between sleep and molecules, as between sleep and its functions and, even more recently, between sleep and sleep disorders with our genes.

1.2 Sleep Studies

Doctors can diagnose abnormal sleep patterns or any other sleep-related problem with the data collected with a sleep study. Although it is usually necessary, the goal of sleep studies is not to characterize the sleep structure but to detect the symptoms and diagnose disorders which may be causing sleep problems and impacting daily life. When referring to disorders it is now common the term dyssomnias, avoiding the use of hypersomnias or insomnias, as poor sleep has a direct consequence in daytime somnolence.

Some common sleep studies are the Multiple Sleep Latency Test, the Actigraphy, and the Pulse Oximetry, described below.

Multiple Sleep Latency Test (MSLT) is the study of the input latency of sleep and REM phase. It is done recording multiple naps during the same day with usually two hours between consecutive naps. This test is used to detect a pathological hypersomnolence and even its relationship with a specific disorder such as narcolepsy.

Actigraphy is a study that evaluates the movement for several days. The usual case is to place an accelerometer in the wrist and record periods lasting between four and ten days. The idea is to estimate sleep periods of patients with sleep problems using the arm's movement.

Pulse Oximetry is a study that monitors oxygen concentration in hemoglobin, usually measuring pulse rate as well. It is used to discard sleep apneas or to control if a treatment is working in already diagnosed patients.

However, the most popular and standard sleep study is polysomnography.

1.2.1 Polysomnography

Polysomnography (PSG) is the most common sleep study, used both to characterize sleep and as a diagnostic tool. It consists of the recording of multiple signals during the night, placing sensors over the patient to monitor their physiological functions. The number of recorded signals, as the number of sensors, depends on the symptoms and their probable diagnose. Some studies only need to record neurophysiological activity, while others require more signals, like those related to the respiratory function.

Polysomnography is used to diagnose, or to rule out, many types of sleep disorders, including narcolepsy, idiopathic hypersomnia, periodic limb movement disorder, REM behavior disorder, parasomnias, and sleep apnea. It is also useful to rule out other sleep disorders and to detect episodes happening during sleep as awakenings, somniloquy, sleepwalking, bruxism, or night terrors.

The standard test is carried out in the sleep laboratories of medical centers. The patient comes to the medical center in the early evening and over the next one or two hours is introduced to the setting, and the sensors are connected to record multiple channels of data when it falls asleep. A sleep technician should always be in attendance and is responsible for attaching the electrodes to the patient and monitoring the patient during the study. Not only the attached sensors but the disruption in the patient's routine makes this procedure uncomfortable, biasing the results as it impacts the patient's sleep. This problem can be limited with new portable devices that offer the possibility of doing the study at home, but their use is usually limited to a screening function and not as reliable as in site monitoring.

The usual PSG montage involves three different types of signals: pulmonological, neurophysiological, and contextual information.

Pulmonological signals

It is the set of signals comprising movements, oxygen saturation in arterial blood and airflow. The two most common are respiratory airflow and oxygen saturation.

- **Respiratory airflow:** is the signal that monitors the volume of air inhaled and expelled from the lungs, resulting in a sinusoidal signal that reflects the respiratory rhythm. The AASM suggests the simultaneous use of a pressure transducer and a thermistor to record the airflow.

- **Oxygen saturation:** is the signal that monitors oxygen saturation in arterial blood in a non-invasive manner, obtaining the percentage of arterial hemoglobin measuring the changes in light absorption resulting from beats in the arterial blood flow.

Neurophysiological signals

It is the set of signals related to the sleep function. The most common are electrooculogram, electromyogram, and electroencephalogram.

- **Electrooculogram (EOG):** is the signal that monitors the ocular movements using the difference in the potential between the cornea and the retina of the eye, characterizing the eye as a rotatory dipole. It helps to distinguish the different patterns of eye movements which occur during some sleep periods.

- **Electromyogram (EMG):** is the signal that monitors the neuromuscular activity associated with muscle contraction, which is recorded using an electrode over the skin surface of the corresponding muscle. Typically, the recorded signals are the submental EMG, because it reflects changes in the normal progression of sleep, and two tibial derivations to track legs movements.

- **Electroencephalogram (EEG)[1]:** is the signal that monitors the brain electrical activity, placing electrodes over multiples areas of the scalp, generally in a bipolar setting where one extreme is attached to a specific region and the other to a reference one. It is the most complex of the neurophysiological signals involved in the characterization of sleep as it is non-linear, non-stationary, and has a low signal-to-noise ratio.

[1] See Appendix A to read more about the EEG, specifically the EEG wave patterns

Contextual information signals

It is the set of signals that are not directly related to the sleep function. This category also includes those that comprise information regarding the context of the study.

- **Body position:** is the signal that monitors the body position using an accelerometer attached to the patient's trunk. The recorded position is usually discretized in four key body positions: supine, prone, left lateral, and right lateral.
- **Lights control:** is the signal that monitors the periods in which the patient is already in the bed and about to sleep. It is used to discard intervals in the PSG in which the recording is active but that are not valid sleep periods.
- **Snore sound signal:** is the signal that monitors the ambient sound, mainly to record snore, as it may be a hint to locate respiratory pauses.
- **Electrocardiogram (ECG):** is the signal that monitors the heart electrical activity. Although it is not used to characterize sleep, it is recorded as it is the vital monitoring signal.

1.3 Sleep structure

Experts can analyze the data collected in a PSG to characterize the sleep structure, which we can divide into two categories: macro and microstructure. The macrostructure is the classification of sleep stages, which is the evolution of the sleep process. The microstructure is the set of different transient events that appear during the distinct sleep stages such as arousals or sleep spindles.

1.3.1 Sleep Stages

As aforementioned, Loomis et al. [14] were the first to observe that sleep is not a homogeneous state, describing different stages based on the EEG. In 1953, Aserinsky and Kleitman observed a particular state of sleep characterized by rapid, binocularly symmetrical eye movements [13]. They named it rapid eye movement (REM) sleep.

The brain activity measured with the EEG during REM is similar to that during wakefulness. Also, both respiratory and heart rate are higher compared to other sleep stages. Studying the overnight recording of EEG and EOG, Kleitman and Demet found a cyclic pattern of REM and non-REM (NREM) sleep [23]. Later, Aserinsky and Kleitman [13] divided NREM sleep into four stages, ranging from the lightest sleep in stage 1 to the deepest sleep in stage 4. Traditionally, the analysis of the sleep structure is done using three primary sources: EOG, EMG, and EEG; standardized since the work of Rechtschaffen and Kales (R&K) [17]. The R&K manual includes parameters, techniques and wave patterns commonly detected in PSG recordings, and it was the first standardization of sleep analysis, lasting until 2007.

The publication of a uniform and standard criteria was a necessity to increase the comparability and replicability of the results from different laboratories. According to the R&K criterion, sleep is divided into two great stages: REM and NREM. They also split NREM into four stages, following the conclusions of Aserinsky and Kleitman. The R&K manual also defines 20 or 30 seconds time windows, namely epochs, and suggests to score sleep stages following an epoch-by-epoch approach. The structural analysis of sleep proposed by R&K was the standard method until the American Academy of Sleep Medicine (AASM) published a modification. The AASM manual [22], published in 2007, was a response to the advances in sleep medicine. Thus, it included new knowledge, technical methods, and capabilities. Given the time-proved validity and reliability of the R&K system, the AASM rules and specifications kept most of the framework, adding new definitions and some rule modifications. Moreover, the AASM publication included conditions for the characterization of pediatric patients and added a set of events into the standard scoring system as arousals, movements, and respiratory and cardiac events. Regarding the sleep macrostructure, the most significant change was the fusion of stages 3 and 4 into a single stage representing deep sleep.

The AASM manual defines a total of five stages: Wakefulness (W), Rapid Eye Movements (REM), and three non-REM stages namely Stage 1 (N1), Stage 2 (N2), and Stage 3 (N3).

Stage W

Stage W represents the waking state, ranging from full alertness to early stages of drowsiness. Electrophysiological and psychophysiological markers of drowsiness may be present during stage W and may persist into stage N1. In stage W, the majority of individuals with closed eyes show alpha rhythm[2]: trains of sinusoidal 8-13 Hz activity recorded over the occipital region which attenuates with eyes opening. The EEG pattern with opened eyes consists of low amplitude activity (beta and alpha frequencies) without the alpha rhythm. During wakefulness, the EOG may demonstrate rapid eye blinks at a rate ranging 0.5-2 Hz. With the progress of drowsiness, the frequency of blinking decreases, and eye blinks may be replaced by slow eye movements, even in the presence of continued alpha rhythm. If the eyes were open, we would see voluntary rapid eye movements or reading eye movements. The muscular activity registered with the chin EMG during stage W is usually higher than during sleep stages.

Stage N1

It is the lightest sleep state in which the subject can still perceive the majority of stimuli which happen around. Sleep in stage N1 is not practically restful at all. In subjects which generate alpha rhythm, stage N1 is scored when the alpha rhythm is attenuated and replaced by low amplitude, mixed frequency (4-7 Hz) activity for more than 50% of the epoch. Other hallmarks of this stage are the presence of vertex sharp waves[3] and slow eye movements. Slow eye movements are characterized by reasonably regular, sinusoidal eye movements with an initial deflection usually lasting more than 500 ms. During stage N1, the muscular activity registered in the chin EMG is variable but often lower than in stage W.

Stage N2

In this stage, our thalamus blocks sensorial inputs, provoking a disconnection from the environment which facilitates the sleeping process. Sleep in this stage is partially recovering, probably not enough to rest entirely. During stage N2, EEG activity

[2] See Appendix A to read more about the EEG wave patterns

[3] This and other events named in this section are explained in Section 1.3.2

shows low amplitudes and mixed frequencies with the predominance of theta frequency waves. It is also characterized by an increase in delta activity, compared to delta activity during stage N1. However, the primary physiological activities during stage N2 are sleep spindles and k-complexes. The EOG usually shows no eye movements, but slow eye movements may persist in some subjects. The muscular activity registered in the chin EMG is variable, but it usually is lower than in stage W or N1.

Stage N3

In this stage, sensorial blocking intensifies compared to stage N2, indicating a deeper sleep. If the subject wakes up in this stage, it will probably suffer confusion and disorientation. Sleep in stage N3 is essential for a restful sleep. The EEG activity shows slow waves with a predominance of delta frequency. Slow wave activity includes waves of frequency 0.5-2 Hz with a peak-to-peak amplitude higher than 75 μV, measured in the frontal regions. Typically, stage N3 is scored when at least 20% of an epoch consists of slow wave activity. Sleep spindles may persist in this stage, but the EOG typically shows no eye movements. The muscular activity registered with chin EMG is often lower than in stage N2 and sometimes as low as in stage REM.

Stage REM

It is the stage when the subject dreams. Cerebral activity in stage REM is high, with low amplitude and mixed frequency, with a predominance of theta activity and the possible presence of beta bursts. Thus, it is similar to the activity that appears during stage N1. In this stage, the typical transient pattern of EEG activity is saw-tooth waves. In some individuals, alpha activity is higher in stage REM than in stage N1. However, alpha frequency in stage REM is often 1-2 Hz slower when compared to wakefulness. Rapid eye movements are characteristic of this phase. We can identify them as conjugate, irregular, sharply peaked eye movements with an initial deflection usually lasting less than 500 ms. Transient muscle activity is also frequent in the EMG, although it usually reaches its lowest amplitude. It appears as short irregular bursts, regularly lasting less than 0.25 s. This activity is maximal in association with rapid eye movements.

1.3.2 Sleep microstructure

The identification of events happening during sleep is essential to facilitate the characterization of sleep macrostructure. Also, these events can point to specific disorders. For example, the number of arousals during the night (arousal index) can be indicative of sleep apnea. Regarding the sleep microstructure, the research presented in this thesis is focused on arousals and sleep spindles, although other events are also useful for the complete characterization of sleep.

Arousals

The AASM defines the electroencephalographic arousal as an abrupt shift in the EEG frequency including alpha, theta, and frequencies higher than 16 Hz (but not spindles), that last at least 3 seconds and with at least 10 seconds of previous stable sleep. Arousals are a response in the form of alert produced during sleep that does not reflect a total awakening of the subject, although most of the times they imply a change from a deeper sleep stage to a lighter one. As an indicator of disrupted sleep, arousals are an excellent quantification of sleep quality.

These events alter standard sleep architecture, and the sleep fragmentation they cause is one of the main reasons for the daytime sleepiness associated with some sleep disorders. For scoring arousals, at least one central derivation of EEG needs to be recorded. Arousal scoring can also incorporate information from the occipital region. During stage REM it is also required a concurrent increase in the submental EMG activity lasting for at least one second.

Sleep Spindles

Sleep spindles are defined as a train of distinct waves with frequency 11-16 Hz (most commonly 12-14 Hz) lasting at least half a second, usually maximal in amplitude using central derivations. They are one of the hallmarks of stage N2 and one of the few EEG events uniquely related to sleep [24]. Berger was the first to describe this event [25], but Loomis et al. [26] named them. This event is a group of rhythmic waves which progressively increase their amplitude and then gradually decrease. They are usually linked to low voltage background EEG, superimposed to delta activity, or happening simultaneously with a vertex sharp wave or a k-complex.

Sleep spindles show intra-cycle variations in the form of U-shape within the first four sleep cycles and their presence increases with consecutive sleep cycles. Spindle amplitude and density decrease with age but the high intra-individual variability make it difficult to asses it.

Other events

- **K-complex:** is a brief negative high-voltage peak, usually higher than 100 μV, followed by a slower positive complex around 350 and 550 ms and at 900 ms a last negative peak.
- **Vertex waves:** are distinctive 'V' shaped waveforms with peaks reaching 100-200 μV and with the largest amplitude in the middle.
- **Sawtooth:** is a train of vertex waves that can appear in stage REM.

1.4 Research of this thesis

We now summarize the articles included in this thesis. As aforementioned, most of our studies describe the design of automatic algorithms for the characterization of sleep. The first two articles describe approaches for the detection of EEG arousals. We validated the second approach in a real environment, and present our findings in another article. The last article regarding sleep microstructure deals with the detection of sleep spindles. To classify sleep stages, we include two additional works, both using deep learning. Finally, the last article included in this thesis is a case study in which we developed an API for increasing the usability of our algorithms.

1.4.1 Detection of EEG arousals

Multiple researchers have already proposed methods for the automatic detection of EEG arousals. Some works do this detection using a single channel, although this a slight simplification of the problem, as the definition of EEG arousal involves both the EEG and EMG signals. Examples following this approach are works that use the peripheral arterial tonometry [27], the heart rate variability [28], or just a single derivation of the EEG signal [29, 30, 31].

Undoubtedly, some research works use multiple signals, analyzing them with wavelets [32] or applying pattern matching [33]. Either way, the most common approach is first a step of feature extraction to build a vector of features, and second the classification of the vector of features using machine learning. The differences between these methods are the set of extracted features or the particular methods used for the classification [34, 35].

Our first work follows this latter approach whereas the second one relies on pattern matching. We also carried out experiments with the second approach in a sleep center, validating our algorithm in a real environment.

Combining machine learning models for the automatic detection of EEG arousals

This method is based on feature extraction and classification, as we outline in Figure 1.1. Firstly, we decide in which epochs it is possible to detect arousals, then we extract the vector of features and classify it to detect if the epoch contains or not an arousal.

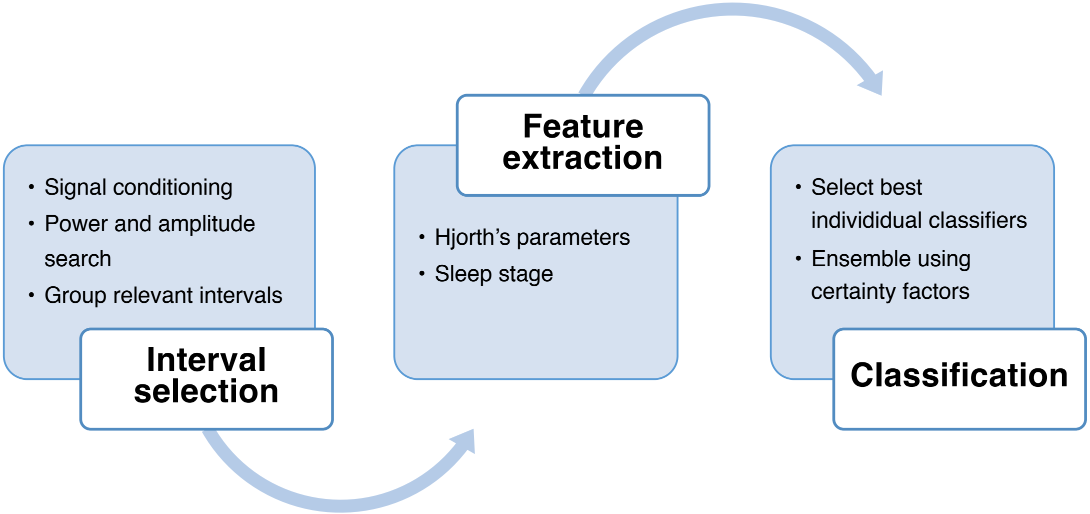

• Signal conditioning
• Power and amplitude search
• Group relevant intervals
Interval selection
Feature extraction
• Hjorth's parameters
• Sleep stage
• Select best individidual classifiers
• Ensemble using certainty factors
Classification

Figure 1.1: Outline of *Combining machine learning models for the automatic detection of EEG arousals*

Our proposal uses four different signals: two EEG derivations, EMG, and ECG. The latter is only used to remove artifacts from the first ones. The heartbeat induces a spike in the EEG signals, introducing noise in the system. To remove the artifacts, we first need to locate the heartbeats finding the QRS complex in the ECG signal. Then, we interpolate the EEG and EMG signals between the limits of each QRS complex to remove the artifacts.

To select the relevant epochs, we search for events in each signal. An event is an abrupt frequency change in the EEG signal and an amplitude change in the EMG signal. We can find these changes comparing the measure of a particular magnitude in a window against the average of measures in the previous windows. In the case of the EEG, we use three magnitudes: the power in the alpha band, theta band, and the power for frequencies higher than 16 Hz. In the case of the EMG, the magnitude is the amplitude of the signal. We consider an epoch as relevant if we find at least one event for each signal.

If an epoch is relevant, we extract features from each of the events to build our vector. The features for the EEG events are the power on the delta, theta, alpha, sigma, and greater than 16 Hz bands, and the Hjorth's parameters [36], namely activity, mobility and complexity. From the EMG, we extract values regarding the amplitude. We complete the vector of features adding the sleep stage and the overlapping time between the EEG intervals.

Using a database of 20 PSG recordings from the Sleep Heart Health Study [37], we created two balanced datasets for training and testing. With the first one, we trained six different classifiers: linear discriminant [38], support vector machine, neural network [39], classification tree [40], k-nearest neighbor, and naïve bayes [41]; using a grid search to configure the best possible hyperparameters for each classifier. We selected the four methods with the highest area under the curve to build an ensemble. The ensemble combined the outputs of the individual classifiers with two different approaches. The first one follows Shortliffe and Buchanan's certainty factors model [42], considering the output of each individual model as a certainty factor. The second approach is a linear combination such that the sum of the weights given to each individual model is 1.

Then, we carried out new experiments using an independent 26 PSG recordings dataset. Our ensemble outperformed not only the individual methods but also well-known ensembles, namely Random Forest [43] and an ensemble of k-NN [44], as Table 1.1 shows. With the certainty factors approach we obtained a sensitivity of 0.78, a specificity of 0.89, and an error of 0.12; with the linear approach a sensitivity of 0.81, a specificity of 0.88, and an error of 0.13.

Classifier	Error	Sensitivity	Specificity	AUC
SVM	0.159	0.845	0.840	0.843
ANN	0.224	0.900	0.754	0.827
CT	0.161	0.785	0.849	0.817
k-NN	0.173	0.814	0.829	0.822
RF	0.160	0.844	0.839	0.842
k-NNE	0.328	0.908	0.629	0.768
S&B Combination	0.124	0.781	0.893	0.837
Linear Combination	0.133	0.810	0.878	0.844

Table 1.1: Results from the combined approaches, the individual and the ensemble models using the 26 PSG recordings dataset.

A simple and robust method for the automatic scoring of EEG arousals in polysomnographic recordings

Although our previous method achieves good results, its inherent problem is the difficulty to explain its decisions. We developed this new method with simplicity in mind. Results can be explained because they are based on physical measures. As Figure 1.2 outlines, this method detects arousals finding their pattern in the relevant signals.

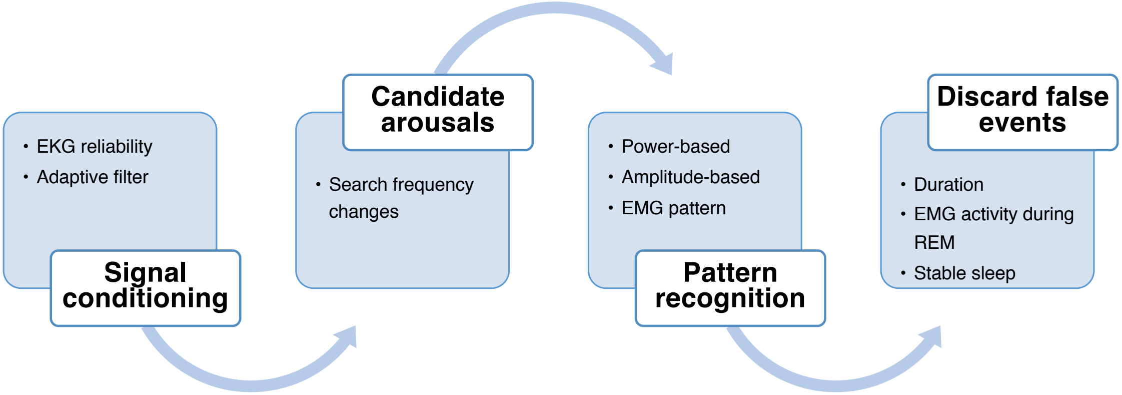


Figure 1.2: Outline of *A simple and robust method for the automatic scoring of EEG arousals in polysomnographic recordings*

As in our previous work, this algorithm starts conditioning the signals. We reduce noise with a high pass and notch filters and remove ECG artifacts from the EEG signal with an adaptive filter. To do this removal, we first study the reliability of the ECG, which allows us to know when it is safe to apply and update the filter. The filter is built and updated using the EEG signal at each heartbeat.

Using the signals without artifacts, we search for frequency changes in the EEG signal both in alpha and beta bands. To do this search, we use a sliding window and compare the power of the current window against the average values of the previous ones. Each frequency change is then studied, trying to recognize an arousal pattern.

We recognize three different patterns. The first one is based on EEG power. As we have already found a power increase, we find when it decreases to the previous values. The second pattern measures the EEG amplitude. It is well known in clinical practice that some arousals also increase the signal amplitude. Thus, for each power increase, we check if the amplitude has also increased. In that case, we check when it goes back to normal. Finally, the third pattern is pretty similar to the former, but measuring the amplitude of the EMG signal.

With the power search, we find a possible beginning for the arousal, whereas recognizing the pattern we find its possible end. It only lasts to check if it is an arousal or some other event. We initially assume it is and then discard it in some cases. The first case depends on the duration, removing those events lasting less than 3 s which is the minimum duration according to the definition, and those lasting more than 15 s, which probably means that for that epoch, stage is W. The second case is to discard it if it is a spindle, which only depends on the primary frequency of the event. We discard those with a main frequency between 12 and 16 Hz. The third one deals with the requirement of EMG activation during REM. If the epoch is in stage REM, but there is no amplitude change in the EMG, we discard the event. Finally, we also discard the event if it is close to a previous one, as in that case there is no 10 s of stable sleep.

We tested the method using 22 PSG recordings from a real patients database, obtained in the sleep center of the Haaglanden Medisch Centrum (HMC) in The Hague, The Netherlands. We obtained encouraging results with a precision value of 0.86 and an F1 score value of 0.79. According to the kappa coefficient obtained comparing our scored arousals against the gold standard (0.78), the agreement is almost perfect. Table 1.2 compares our results against those reported in other works. It also includes the scoring unit, as not all works evaluate the classification of arousals the same way.

Method	#Recordings	Scoring unit	Sensitivity	Specificity	AUC	Precision	F_1 score	Kappa
Pacheco and Vaz [34]	8 (2 hours)	30	0.88	-	-	-	-	-
Cho et al. [30]	6 (NREM)	1	0.75	0.93	0.84*	-	-	-
Sugi et al. [45]	8	1.28 (30 for TN)	0.82	0.88	0.85*	-	-	-
Shmiel et al. [46]	20	30	0.75	-	-	0.77	-	-
Alvarez-Estevez and Moret-Bonillo [47]	5	30	0.86	0.77	0.82	0.42*	0.57*	0.44*
Alvarez-Estevez et al. [48]	26	30	0.65	0.95	0.80	0.7*	0.68*	0.62*
Fernández-Varela et al. [49]	26	30	0.81	0.88	0.85	0.56*	0.66*	0.58*
Ours	22	30	0.75	0.99	0.87	0.86	0.80	0.78

Table 1.2: Results reported for methods using a fixed time window to compute agreements and disagreements against the clinical reference; * Values were not explicitly mentioned in the referenced work, but can be derived from the published data; AUC = Area Under ROC Curve of one point obtained as (Sensitivity+Specificity)/2; NREM = Non-Rapid Eye Movement; TN = True Negative.

Large-scale validation of an automatic EEG arousal detection algorithm using different heterogeneous databases

So far, experiments with our methods were carried out using a single database. In this work, we used two different sources of PSG recordings. The first one is the Sleep Heart Health Study database and the second one is a database containing private recordings from the Haaglanden Medisch Centrum (HMC) in The Hague, The Netherlands. With both sources, we built three datasets: SHHS2, HMC-S, and HMC-M; described in Table 1.3.

Dataset	n	Age	Gender	ArI	AHI
HMC-S	220	52.99 ± 14.33	62% M/58% F	12.85 ± 07.86	13.33 ± 15.28
HMC-M	252	51.58 ± 16.22	53% M/47% F	12.45 ± 10.48	14.83 ± 21.03
SHHS2	2296	67.41 ± 10.03	45% M/55% F	12.91 ± 07.02	16.25 ± 15.64

Table 1.3: Summary of demographic data and main PSG characteristics for the different datasets. n = number of recordings, M = males, F = females, ArI = arousal index, AHI = apnea-hypopnea index

The code used to detect arousals is an adaptation of our previous work (1.4.1). We made changes to support different montage configurations, signal sampling rates, and filtering. We also updated the thresholds, to increase the capacity of detecting alpha frequency changes and to improve how we discard events for being sleep spindles. Finally, we simplified the process, removing some steps when dealing with alpha events.

We evaluated the algorithms with two complementary approaches. Firstly, an event-to-event scoring validation using 30 s epochs, obtaining the results shown in Table 1.4. Secondly, using the Arousal Index (ArI) calculated for each recording. We obtained correlation coefficients among the respective automatic and clinical reference ArI scores. We used the Wilcoxon signed rank to obtain statistical significance for paired differences, and also the Intraclass Correlation Coefficient (ICC) [50] as a measure of repeatability to examine scoring differences. Table 1.5 shows the mean ArI (with deviation) for each dataset and the mean difference between our method and the gold standard. Results reject the null hypothesis H0 "median of differences is zero" ($\alpha = 0.05$), but for the HMC-M dataset. Although differences for HMC-S and SHHS2 are not significant if we assume a median difference bias of 0.3.

Dataset	#Epochs	Sensitivity	Specificity	Precision	F1-score	Kappa
HMC-S	207312	0.580	0.972	0.707	0.637	0.600
HMC-M	236336	0.563	0.953	0.641	0.600	0.559
SHHS2	2201487	0.517	0.979	0.743	0.610	0.573

Table 1.4: Overall results of the event-by-event epoch-based validation on the testing datasets.

Dataset	Gold standard	Method	Difference	p-value
HMC-S	13.32 ± 08.01	12.47 ± 08.06	0.84 ± 5.41	0.023
HMC-M	12.45 ± 10.48	12.97 ± 10.14	0.52 ± 6.68	0.224
SHHS2	12.91 ± 07.02	12.56 ± 07.73	0.35 ± 4.89	< 0.001

Table 1.5: Mean ArI score for each dataset, including the mean difference between them, and p-value for the hypothesis: "mean difference is zero".

In this work, we also assess inter-scorer reliability. An independent expert scored a set of representative recordings. We selected these recordings according to the previous kappa index in the individual event validation. Table 1.6 shows the average kappa index for each dataset, comparing the new expert (Rescoring), the original expert (Original), and our method (Method). These results show that our method performance is similar to what we would expect from another expert.

Dataset	Rescoring vs Original	Method vs Original	Method vs Rescoring
HMC-S	0.594	0.595	0.602
HMC-M	0.561	0.523	0.686
SHHS2	0.543	0.552	0.564

Table 1.6: Average kappa index comparing the experts and our method for the different datasets.

1.4.2 Detection of Sleep Spindles

The detection of sleep spindles only requires the EEG signal. Thus, some works only used band-pass filters and amplitude detection [51, 52], or the teager energy operator and thresholds [53]. Many other works follow the same approach we aforementioned for the detection of arousals, consisting of a first step of feature extraction and a subsequent one of classifying a vector of features. Again, the differences appear in the feature extraction or the classification methods. For feature extraction, we can find works using the fourier fast transform [54] or adaptive autoregressive modeling [55], between others. To classify, they have used multi-layer perceptron [56] or support vector machines [55].

Our work also follows a two-step approach, decomposing the signal to extract features and classifying them afterward.

A comparison of performance of sleep spindle classification methods using wavelets

As aforementioned, we first extract features from our samples, using wavelet decomposition. Then, we compare the performance obtained using different classifiers. Figure 1.3 outlines our method.

The data used in this work belongs to the Sleep Laboratory of the André Vésale Hospital in Belgium Devuyst et al. [57]. It contains eight segments of 30 minutes from eight PSG from different patients. An expert analyzed them to detect sleep spindles using the central EEG derivation, scoring a total of 289 spindles. Six recordings have a sampling rate of 200 Hz, one of 100 Hz and the remaining one of 50 Hz.

From the available data, we built a dataset containing all the spindles and an

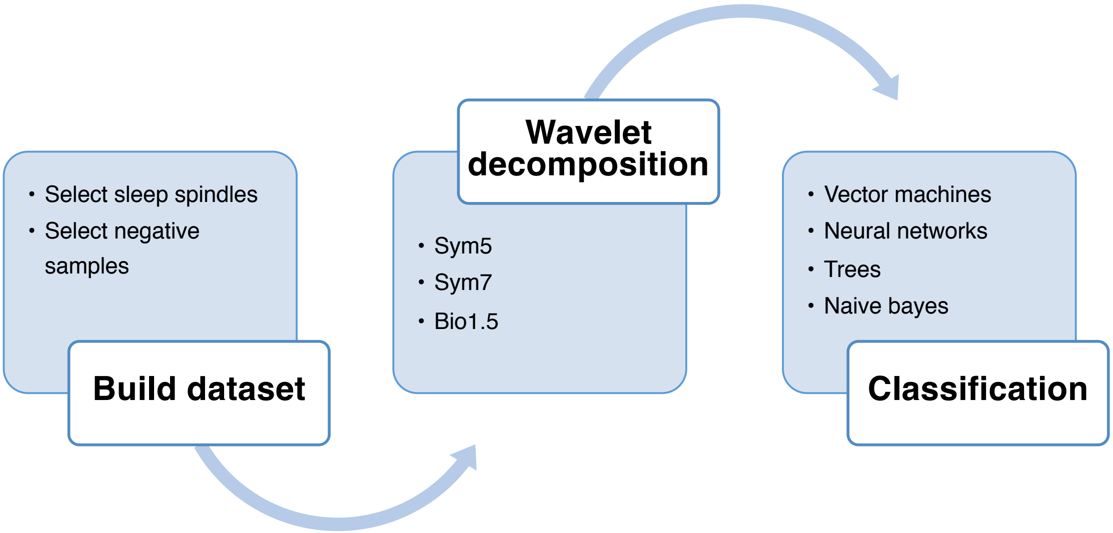


Figure 1.3: Outline of *A comparison of perfomance of sleep spindle classification using wavelets*

equal number of negative samples. Negative samples are randomly selected windows of 0.5 s, which is the minimum spindle duration. We applied wavelet transformation [58] to the samples, obtaining a vector of features from each. We used the symlet of order O5 (sym5) and O7 (sym7), and the biorthogonal of order O1.5 (bio1.5) to decompose each sample and obtain a set of coefficients representing it. Given that we have different sampling rates and even samples of different length, we limited our vector of features to the first 13 coefficients, which would be the minimum number of coefficients in the worst case.

These vectors were then classified using several methods: support vector machine (SVM), proximal SVM (pSVM) [59], feed-forward neural network with one (1 FNN) and two layers (2 FNN), classification tree (CT), random forest (RF) and naive bayes (NB). Tables 1.7 and 1.8 show the accuracy and sensitivity obtained with a 10-fold cross validation with each classifier and wavelet family.

	SVM	pSVM	1 FNN	2 FNN	CT	RF	NB
sym5	89.01 ± 4.3	86.20 ± 3.9	89.58 ± 3.9	87.89 ± 4.5	91.83 ± 4.5	94.08 ± 2.8	93.66 ± 1.7
sym7	86.90 ± 4.5	83.66 ± 6.8	85.35 ± 4.6	82.54 ± 6.6	86.20 ± 3.2	89.58 ± 3.2	85.77 ± 5.2
bio1.5	88.59 ± 3.9	86.20 ± 2.8	88.59 ± 2.7	86.48 ± 6.4	93.38 ± 2.4	94.08 ± 2.4	93.66 ± 2.9

Table 1.7: Mean test set accuracy for the 10-fold cross validation.

	SVM	pSVM	1 FNN	2 FNN	CT	RF	NB
sym5	92.67 ± 4.3	97.99 ± 1.9	96.01 ± 3.4	86.45 ± 6.1	91.86 ± 6.8	95.78 ± 2.8	95.15 ± 1.5
sym7	89.67 ± 4.5	95.04 ± 4.9	90.30 ± 3.8	82.16 ± 7.9	85.47 ± 7.8	88.69 ± 6.5	89.63 ± 4.3
bio1.5	92.08 ± 4.1	99.72 ± 0.8	98.89 ± 1.1	87.05 ± 7.9	93.14 ± 4.3	95.81 ± 3.3	96.36 ± 3.6

Table 1.8: Mean test set sensitivity for the 10-fold cross validation.

1.4.3 Classification of Sleep Stages

The classification of sleep stages is by far the problem in sleep medicine that needs more data. It is also the most explored one. Many works follow the approach we commented regarding the detection of sleep micro-events: feature extraction and classification. Fraiwan et al. [60] used a random forest for the classification of time-frequency features and Renyi's entropy; Liang et al. [61] extracted multiscale entropy and autoregressive features and then applied a linear discriminant analysis; Zhu et al. [62] used features from a difference visibility graph and classified using a support vector machine. Hassan and Bhuiyan [63] followed a single signal approach with wavelet decomposition for the feature extraction and a random forest classifier. The same authors, in other work [64], used finite sums to decompose the signal and compared several classifiers; Sharma et al. [65] studied a discrete energy separation algorithm over a single-channel EEG using iterative filtering, also comparing several classifiers. Koley and Dey [66] applied a support vector machine to frequency, time, and non-linear features extracted from a single-channel EEG. Lajnef et al. [67] used multiple signals and multiple support vector machines to build a decision tree. Huang et al. [68] studied the power spectral density of two EEG channels to obtain frequency-domain features and classified them with a modification of a support vector machine; Finally, Günes et al. [69] used spectral analysis to extract the features and a nearest neighbors algorithm for the classification.

Approaching the classification of sleep stages with feature extraction is inherently biased towards the available dataset, a problem caused by the human engineered features. Thus, most of these proposals cannot generalize to other datasets. Especially, given the nature of PSG recordings, with significant natural differences between individuals apart from those introduced by the recording hardware.

One option to solve this issue is to use a method that can learn from raw data, limiting human bias. Nowadays, the natural choice is deep learning, as it has outperformed previous methods in several fields, particularly in medical diagnosis [70, 71].

Some works have already explored different algorithms that would belong to the deep learning area: Längkvist et al. [72] used deep belief networks to learn probabilistic representations from preprocessed raw signals; Tsinalis et al. [73] studied convolutional neural networks to extract time-invariant features from a raw EEG channel. The same authors also studied the use of stacked sparse autoencoders [74]; Supratak et al. [75] used a convolutional neural network complemented with a bidirectional long short-term memory network (LSTM). Biswal et al. [76] compared a recurrent neural network against other networks but using features as the input. Finally, Sors et al. [77] also used a convolutional neural network with a single channel raw EEG.

Sleep staging with deep learning: a convolutional model

In this first work using deep learning, we simplified the problem, merging stages N1 and N2 into drowsy sleep (DS). This way we avoid the problem of detecting stage N1, the most underrepresented and with the lowest agreement between experts. Our neural network was developed and tested using 240 PSG recordings from the SHHS dataset: 180 for training, 20 for validation and 40 for testing. As expected, the dataset is highly unbalanced. In the training dataset, 39.7% of the samples are classified as W, 38.3% as DS, 9.6% as N3, and 12.4% as REM. In the validation dataset, the distribution is 42.0% for W, 37.3% for DS, 9.1% for N3, and 11.6% for REM. Finally, in the test dataset, the distribution is 42.7% for W, 37.3% for DS, 8.8% for N3, and 11.2% for REM.

The input for the neural network is an epoch, and the output is the predicted sleep stage. Our input contains the five available signals in the SHHS recordings: both EEG channels, the EMG, and both EOG. As they are sampled at different rates, we padded with zeros the signals with lower rates.

We limited our convolutions to one dimension, avoiding an artificial spatial relationship between the signals. We selected the architecture and hyperparameters of the network using the validation set and trying to reduce the number of layers to the minimum. In the end, the network was composed by two convolutional layers each with 128 kernels, one pool layer, another convolutional layer with 256 kernels, a max pool layer, and a last fully connected layer. The filter size was fixed at 20 for every convolutional layer, with padding adjusted to maintain the input dimension. The gradient optimizer was Adam [78] and the activation functions for all the convolutional layers relu [79], except for a final softmax function. We improved regularization normalizing the signals to mean 0 and deviation 1 and adding a dropout [78] in the final layer. The training was done with batches of size 32 and finished using early stopping over the validation loss with a patience value of 3. Table 1.9 shows the results obtained with the test dataset.

Sleep Stage	Precision	Recall	F-1 Score
Awake (W)	0.96	0.96	0.96
Drowsy Sleep (DS)	0.90	0.91	0.90
Deep Sleep (N3)	0.89	0.82	0.85
REM	0.89	0.90	0.90
Average	0.91	0.90	0.90

Table 1.9: Precision, recall, and F-1 score for the classification of the test dataset.

Studying the results for the individual recordings we learned that for stage W the network is robust, always achieving high values. On the contrary, for stage N3 results vary significantly with the recording. When compared against other works (Table 1.10) results are encouraging, although we need to distinguish between stage N1 and N2.

	This work	Alvarez-Estevez et al. [48]	Sors et al. [77]
Awake	0.96	0.88	0.91
Drowsy Sleep	0.91	0.81	0.35 (N1) 0.89 (N2)
Deep Sleep	0.82	0.75	0.85
REM	0.90	0.84	0.86
Average	0.90	0.82	0.88*

Table 1.10: Comparison of our results against previous works reporting recall. * Taking 0.89 as reference for DS

A convolutional network for the classification of sleep stages

The goal of this work is to overcome the limitations of our previous one. Mainly, we classify the five stages recognized by the AASM and improve the selection of hyperparameters. We explain next these and some other changes.

The dataset is also composed of PSG recordings from the SHHS database. In this case, we filtered all the signals, to reduce noise and remove artifacts. The filtering pipeline was already explained in Section 1.4.1. For training, we used 400 recordings, for validation 100 and for testing 500.

The input to the network is an epoch, upsampling the required signals to 125 Hz, which is the highest frequency in our dataset. In this case, we avoid padding with zeros because it would be less generalizable if we try to apply it to other databases. We also avoided downsampling to keep all the frequencies that are relevant for the classification of sleep stages. The neural network is a stack of convolutional blocks. Each of this convolutional block contains a 1D convolution with batch normalization [80] and relu activation and an average pool that reduces the input size by half. The difference between a block and the following one is that the number of filters of the latter is twice the number of filters of the former. Undoubtedly, the input size for each block is half the previous one. Finally, the network includes a global pool with dropout and a dense layer with softmax activation to output the classification probabilities for each class.

As aforementioned, we also improved the selection of hyperparameters. As hyperparameters, we considered the number of convolutional blocks, the size of the kernel, the number of filters of the first block, and the learning rate. To select their values, we used a Tree-structure Parzen Estimator (TPE) [81]. TPE is a sequential based optimization that builds models in sequence trying to approximate the performance of a selection of hyperparameters based on historical results, and then chooses new values that are checked with the model. Using TPE, we trained 50 models to obtain the best possible hyperparameters. Each of these models was trained using a subset of 250 recordings from the original training dataset.

Finally, we selected the five sets of hyperparameters obtaining the best results and built an ensemble with them. We show in Table 1.11 the results obtained with this ensemble when classifying our test dataset (500 recordings).

The class for which we achieved the best classification is W, with values near to 0.95 for the precision, sensitivity, and F1 score. Then, classes N2, N3, and REM showed similar results, especially if we compare the F1 score, although sensitivity for N3 was lower (thus, precision was higher). Lastly, results regarding the classification of class N1 were rather low, not even achieving an F1 score of 0.3.

Stage	Precision	Sensitivity	F1 score
W	0,94	0,96	0,95
N1	0,39	0,21	0,27
N2	0,87	0,89	0,88
N3	0,92	0,77	0,84
REM	0,82	0,90	0,86
Average	0,78	0,75	0,76

Table 1.11: Performance measures for the classification of the test dataset using the ensemble with the 5 selected models.

Table 1.12 shows the results reported in other works compared to ours. We improve the general classification, as shown by the kappa value. The improvement is because we classify better the most common stage, W. Although the F1 score for N2, N3, and REM is also high, and between the best values reported, our method struggles to classify stage N1.

Work	**Database**	**Kappa**	**F1 score**				
			W	**N1**	**N2**	**N3**	**REM**
Biswal et al. [76]	Massachusetts General Hospital, 1000 recordings	0,77	0,81	**0,70**	0,77	0,83	**0,92**
Längkvist et al. [72]	St Vicent's University Hospital, 25 recordings	0,63	0,73	0,44	0,65	**0,86**	0,80
Sors et al. [77]	SHHS, 1730 recordings	0,81	0,91	0,43	0,88	0,85	0,85
Supratak et al. [75]	MASS dataset, 62 recordings	0,80	0,87	0,60	**0,90**	0,82	0,89
Supratak et al. [75]	SleepEDF, 20 recordings	0,76	0,85	0,47	0,86	0,85	0,82
Tsinalis et al. [73]	SleepEDF, 39 recordings	0,71	0,72	0,47	0,85	0,84	0,81
Tsinalis et al. [74]	SleepEDF, 39 recordings	0,66	0,67	0,44	0,81	0,85	0,76
This work	SHHS, 500 recordings	**0,83**	**0,95**	0,27	0,88	0,84	0,86

Table 1.12: Comparison of our results against previous works.

1.4.4 API for Sleep Medicine

One reason for the limited adoption of automatic algorithms in sleep centers is the difficulty to integrate them with other existing software. An option to ease the necessary effort is to provide an application programming interface (API). The goal of an API is to facilitate the use of sophisticated methods giving simple function definitions. This way, complexity is encapsulated and the methods can be easily used.

A systematic approach to API usability: taxonomy-derived criteria and a case study

In this work, we built a set of heuristics and guidelines for API usability that synthesize previous API usability studies and cover missed points. We used the heuristics and guidelines to build an API for sleep medicine. We were able to identify problems in our API that we would not have found following previous works.

When building our heuristics and guidelines, we followed the taxonomy proposed by Alonso-Ríos et al. [82]. We reviewed the previous works regarding API usability mapping the different items described in them with the categories defined in the taxonomy. All items found could be mapped to at least one category. Studying the literature and following the taxonomy we built new heuristics and guidelines. We took some heuristics from the previous studies, others are a synthesis from multiple authors, and the remaining ones cover the categories of the taxonomy that we did not found in the literature.

The requirements of the API for sleep medicine were defined by the usability engineers and API developers, after analyzing the context of use. Analyzing the context of use is essential because it is what distinguishes this particular API from others and even from another type of software development. Then, we developed the first API to use our algorithms, which we submitted to the heuristic evaluation.

The heuristic evaluation was carried out by the usability engineers, using the proposed API, the requirements and the heuristics and guidelines. The result of this evaluation was a list of weak and strong points regarding the usability of the API and a set of proposals for improvement. Each guideline and heuristic can be completely, partially or not fulfilled. Figure 1.4 shows the results of the first evaluation.

With the evaluation we found problems as names that are not self-explanatory, it is not easy to understand what the code does, or errors do not provide helpful information; between others. Most of the problems were addressed and fix, developing a new version of our API.

This last was submitted for subjective analysis, asking the API users to respond to a questionnaire. The API user was a computer scientist from the Haaglanden Medisch Centrum (HMC) in The Hague, The Netherlands. The API user also had an interview with the API developer to clarify the questionnaire. The user could

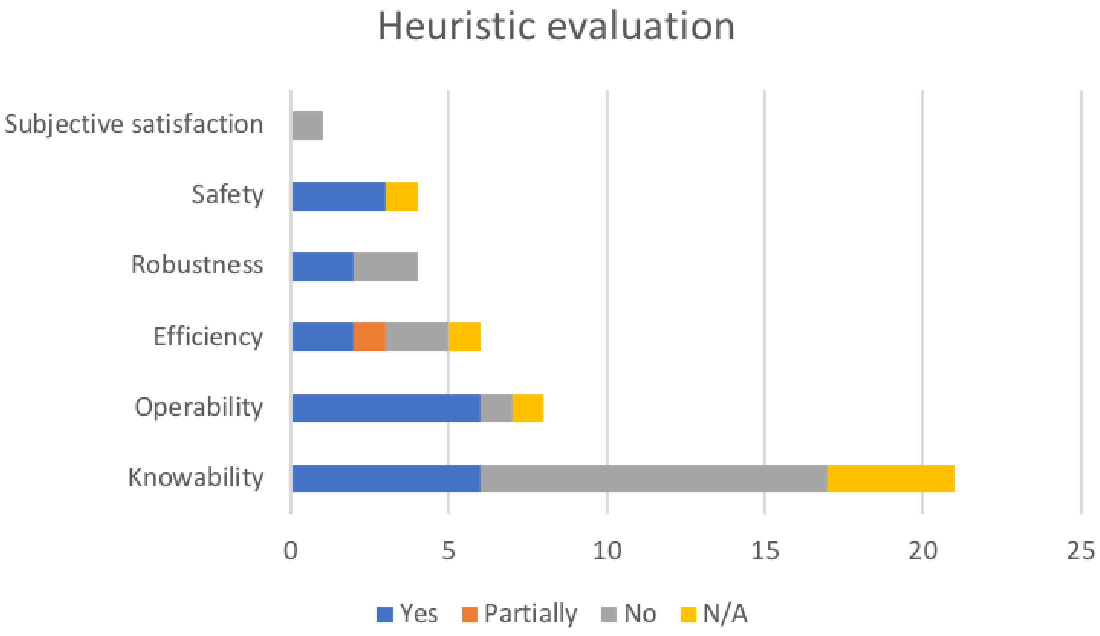


Figure 1.4: Results of the heuristic evaluation

not answer some questions as it did not have access to the source code nor had used the API extensively. Figure 1.5 shows the results of the questionnaire and how the new API version solved most of the problems of the previous one.

We can see that there are still areas for improvement, for example, trying to identify and reduce heuristics that the user classified as “Partially” or developing successive rounds of subjective analysis with new users, trying to involve them in the design of the API.

1.5 Conclusions

Sleep medicine could greatly benefit from methods that can analyze data from sleep studies characterizing the sleep macro and microstructure. If we could automate this time-consuming task, specialists from sleep centers could focus their time on diagnosing the disorder and planning the treatment. Automatic methods would also help to avoid the problems that can appear when human experts are doing the analysis. These are entirely objective and regular methods, always yielding the same result for the same input. In this thesis, we present several techniques that analyze

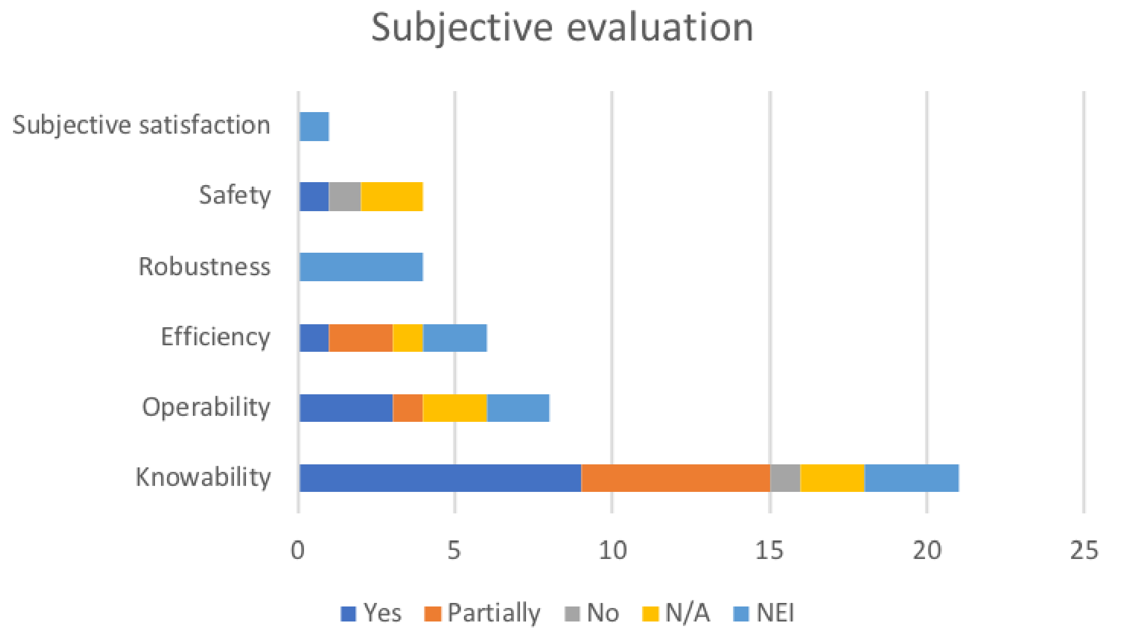


Figure 1.5: Results of the subjective evaluation. NEI = not enough information.

data from sleep studies.

To characterize sleep microstructure, we presented algorithms for the detection of arousals and sleep spindles. Regarding the detection of arousals, our first method relied on feature extraction with a subsequent classification using an ensemble of different classifiers. We concluded that using the sleep stage and the Hjorth's parameters as features improves detection. We improved detection even further using an ensemble, combining the individual predictions following Shortlife and Buchanan method for certainty factors. With our combination of features and the ensemble, we achieved higher sensitivity and specificity than previous methods. The second method finds relevant segments using frequency analysis and then detects arousals studying different patterns. Although it was designed to be a simple method, results showed higher F1 score and kappa index than previous works.

The latter method for the detection of arousals was generalized to use it with more than one database. A common problem of previous studies in the literature is the lack of adaptability. Researchers usually design for a single database and the performance of their methods drops when exposed to other sources of PSG recordings. We showed how our method performs as one more expert when confronted with different databases, achieving an agreement with an expert similar to the agreement between two experts. We carried out our experiments in a real environment, a sleep center from The Netherlands, so we could also demonstrate how automatic algorithms can reduce the time used in sleep characterization.

We also develop an algorithm for the detection of sleep spindles using feature extraction and machine learning. We showed that signal analysis using wavelets could obtain similar results to other proposed solutions in the literature when the classifier is a random forest.

A different approach was used to classify sleep stages. Methods using feature extraction as the first step are usually biased towards a single database. They also depend on the human that engineered the features. To avoid these problems, we relied on methods that can self-learn the relevant features. Our first study, using deep learning and a simplified sleep staging classification problem, showed that this is a valid approximation. The second work we have presented improves our first approach. We still trained a convolutional network, feeding it with the raw epochs from PSG recordings, but considering the five known sleep stages. A fundamental step when developing neural networks is the selection of hyperparameters. We resolved this issue training several models with different configurations. The configuration for a particular model depended on the results obtained with the previous ones. The models with the five best hyperparameters configuration were selected to build an ensemble. The classification achieved with this ensemble outperforms previous works for some sleep stages while obtaining similar results for the remaining ones.

Finally, to facilitate the use of the developed algorithms, we built an application programming interface (API). We improved our first API design following the proposed heuristics and guidelines. Then, we used a questionnaire to interview an engineer and assess the API from its subjective point of view. The engineer's response showed how our redesign API fixed most of the problems encountered with the heuristics and guidelines and yet outline points that could still be improved.

1.6 Future Work

In this thesis, we presented a method that was validated and exploited in a sleep center, which is not common. A possible explanation is that this method is simple enough to earn the trust of experts. Our algorithm makes decisions based on measures from the signal, so the outcome is easily understandable. On the contrary, most automatic methods are not trustworthy. Usually, we present these methods as a black box, with not a single explanation regarding their results. The first proposed line for future work will address this problem. We should explain the results that rely on machine learning or deep learning methods. Explainability is one of the most active areas of research in the field of artificial intelligence.

The other line of future work is the improvement of the presented algorithms. In this sense, we believe that the most promising algorithms are the ones using deep learning. Within this line, we have four new proposals. Some try to address different questions raised while developing our methods, while others try to integrate the sleep characterization.

The first proposal is the detection of sleep micro-events using deep learning, which is more complicated than sleep stages classification, given that this is a time series problem. Also, if we develop new deep learning methods, we should also obtain their saliency maps, a standard way of understanding how the network is detecting events.

The second proposal is to study which is the best input for our networks. So far, we always trained the neural networks using the electric signals recorded in PSG. The most advanced methods in deep learning are those for image classification. Moreover, expert analysis is also based on the image of a single epoch. Thus, we should study networks using images as input.

The third proposal addresses the fact that sleep stages classification also depends on previous stages. So far, none of our methods included memory, which means we cannot take advantage of this property. Thus, we should study networks that remember how they classified previous epochs such as Long Short Time Memory networks (LSTM).

Finally, we should develop methods that achieve excellent results with different databases. To do so, we should know if it is better to do transfer learning or train several models with several databases.

Chapter 2

Detection of EEG Arousals

2.1 Combining machine learning models for the automatic detection of EEG arousals

- **Title:** Combining machine learning models for the automatic detection of EEG arousals
- **Authors:** Fernández-Varela, Isaac, Hernández-Pereira, Elena, Alvarez-Estevez, Diego, Moret-Bonillo, Vicente
- **Journal:** Neurocomputing
- **Editorial:** Elsevier
- **ISSN:** 0925-2312
- **Year:** 2017
- **Volume:** 268
- **Pages:** 100-108
- **DOI:** 10.1016/J.NEUCOM.2016.11.086
- **Available in:** `https://www.sciencedirect.com/science/article/pii/S0925231217307506`

Neurocomputing 268 (2017) 100–108

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Combining machine learning models for the automatic detection of EEG arousals

Isaac Fernández-Varela[a,*], Elena Hernández-Pereira[a], Diego Álvarez-Estévez[b], Vicente Moret-Bonillo[a]

[a] Universidade da Coruña, Departamento de Computación, Facultade de Informática, Campus de Elviña, A Coruña, Spain
[b] Sleep Center, Haaglanden Medisch Centrum, The Hague, Netherlands

ARTICLE INFO

Article history:
Received 6 July 2016
Revised 3 October 2016
Accepted 20 November 2016
Available online 2 May 2017

Keywords:
EEG arousals
Signal processing
Machine learning

ABSTRACT

Electroencephalographic (EEG) arousals are related to sleep fragmentation and the consequent daytime sleepiness, and are usually detected by visual inspection of sleep polysomnographic (PSG) recordings. As this is a time-consuming task, automatic processes are required. A method using signal processing and machine learning models is presented. Using signal processing techniques, after a first step of signal conditioning, abrupt frequency changes in two EEG derivations and amplitude events in one submental electromyogram are identified. These events are grouped if they occur at the same time, using the epoch segmentation for that purpose. A set of features (that includes Hjorth's Parameters and the Sleep Stage), is extracted from each group and used as input for several machine learning models. With a first dataset of 20 PSG recordings, six models are configured and compared: Fisher's Linear Discriminant, Support Vector Machines, Artificial Neural Networks, Classification Trees, k-Nearest Neighbors, and Naive Bayes. The best models, in terms of the classification error and the capabilities to detect EEG arousals, were used to build two different combined approaches. The first approach follows the Shortliffe and Buchanan's certainty factors model and the second follows a linear combination. Conducting experiments on 26 PSG recordings, a sensitivity of 0.78 and a specificity of 0.89 with an error of 0.12 was achieved using the first approach, and a sensitivity of 0.81 and a specificity of 0.88 with an error of 0.13 was achieved using the second approach. Both approaches improved the performance over the individual models. These results were also compared to two well-known ensemble methods: Random Forest and k-Nearest Neighbor Ensemble. Again, the combined approaches showed the best performance.

1. Introduction

The American Academy of Sleep Medicine (AASM) defines the electroencephalographic arousal as an abrupt shift in electroencephalogram (EEG) frequency, including alpha, theta, and/or frequencies greater than 16 Hz, lasting at least 3 s and with at least 10 s of previous stable sleep [1]. Furthermore, during the rapid eye movement (REM) phase, a concurrent increase in the submental electromyography (EMG), lasting at least 1 s, is needed to score such an event. EEG arousals alter the normal sleeping pattern, causing fragmented sleep, the second most common disease indicator, after pain. A high number of EEG arousals during sleep are related to daytime sleepiness. Thus, sleep studies must identify these events for a correct diagnosis.

Usually, sleep studies are performed with an overnight test, called polysomnography (PSG), which is the standard for the diagnosis of multiple disorders [2]. The goal of this procedure is to record a set of physiological signals from the patient, including pneumological signals, electrophysiological signals, and other contextual information. An expert physician examines the signals, detecting different events throughout the recording. Following the standard procedure, to find an EEG arousal, at least one central derivation of EEG and the EMG needs to be recorded. Once individual events are located and associated with their occurrence time, clinical evidence patterns can be formed. Each pattern provides information, both from the individual events and from their structure, which allows the expert to decide about the presence of an arousal event during the pattern interval. Since the recording of a PSG lasts a whole night, the amount of data is huge, making the detection of EEG arousals a very time-consuming task. Thus, automatic detection and analysis are desirable.

[*] Corresponding author.
E-mail address: isaac.fvarela@udc.es (I. Fernández-Varela).

http://dx.doi.org/10.1016/j.neucom.2016.11.086
0925-2312/© 2017 Elsevier B.V. All rights reserved.

Several previous works have attempted to achieve the goal of solving this automatic detection problem with differing levels of success. Even though the standard procedure implies the use of at least one EEG derivation and one EMG derivation, some authors have proposed the use of other signals. For example, Pillar et al. [3] studied the problem using a peripheral arterial tonometry (PAT) to obtain the arousal index, using the PAT amplitude and the changes in the pulse rate. Telser et al. [4] followed a similar idea, detecting EEG arousals using the heart rate variability. Other works, such as Gouveia et al. [5] or Cho et al. [6] only used information from the EEG, avoiding REM stage identification. Agarwal [7] also only used EEG derivations, but in this case two derivations, one for the study of the power in the alpha frequency band and the other in the beta band. The use of the standard procedure is a more complex task, including the difficulty of analyzing those events occurring in different signals at the same time. In this context, De Carli et al. [8] proposed a method using wavelets for the study of the EEG and average measures for the study of the EMG, while Malinowska et al. [9] applied pattern recognition techniques in the EEG and studied the deviation of the EMG signal during REM stages. Other works followed a machine learning approach after a first phase of signal processing. With this technique, Pacheco and Vaz [10] used a K-means classifier after obtaining the frequency and power of the EEG and EMG respectively, and Alvarez-Estevez and Moret-Bonillo [11] compared different classification models after selecting intervals based on the frequency from two EEG derivations, and on the amplitude from one submental EMG. A similar approach was followed by Shahrbabaki et al. [12], but including information from leg movement, airflow and electrocardiography (EKG). Finally, the work of Wallant et al. [13] avoided the use of machine learning algorithms, scoring the EEG arousals after searching for abrupt frequency changes in the EEG and for muscular activation in the EMG.

The method described in this paper uses three signals (plus another one for conditioning purposes) analyzing their relevant parameters: the power in selected frequency bands in two EEG derivations, and the amplitude in an EMG signal. From these analyses, we obtain a set of intervals that must be related to one another, following temporal constraints. Each group of intervals is used to extract multiple features that are the input to several machine learning models. The conducted study includes six different models: Fisher's Linear Discriminant, Support Vector Machine, Artificial Neural Network, Classification Tree, k-Nearest Neighbor, and Naive Bayes. At the end, the outputs of these models are combined to obtain a final decision. Two different approaches were tried to achieve this combination, one following the Shortliffe and Buchanan's certainty factor models and the other one following a linear combination. The output of these approaches was compared to the output of two well-known ensemble methods: Random Forest and k-Nearest Neighbor Ensemble.

With this work, we propose new algorithms for the selection of relevant intervals from the input signals, the use of a set of features not common in the literature, including Hjorth's parameters and sleep stage, and, finally, two different combined approaches, built after selecting the best individual models, demonstrating that they improve the performance of the individual and ensemble models.

2. Proposed method

The proposed method works in a multichannel context, analyzing three signals: two EEG derivations (C3/A2 and C4/A1) and one EMG (submental); making the final decision on the presence of EEG arousals with machine learning models. In the first step, a fourth signal, the electrocardiogram (EKG), is used to remove artifacts from the EEG derivations. Before applying the machine learning models, using signal processing techniques, the suitable intervals from the different signals are selected and grouped to extract defining features from them. Fig. 1 represents the structure of the method proposed in this paper.

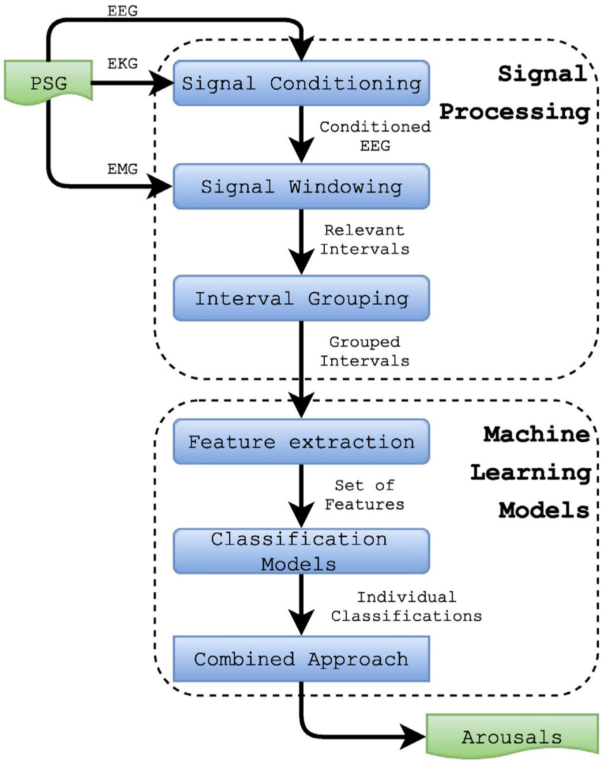


Fig. 1. Structure of the method proposed.

2.1. Signal processing

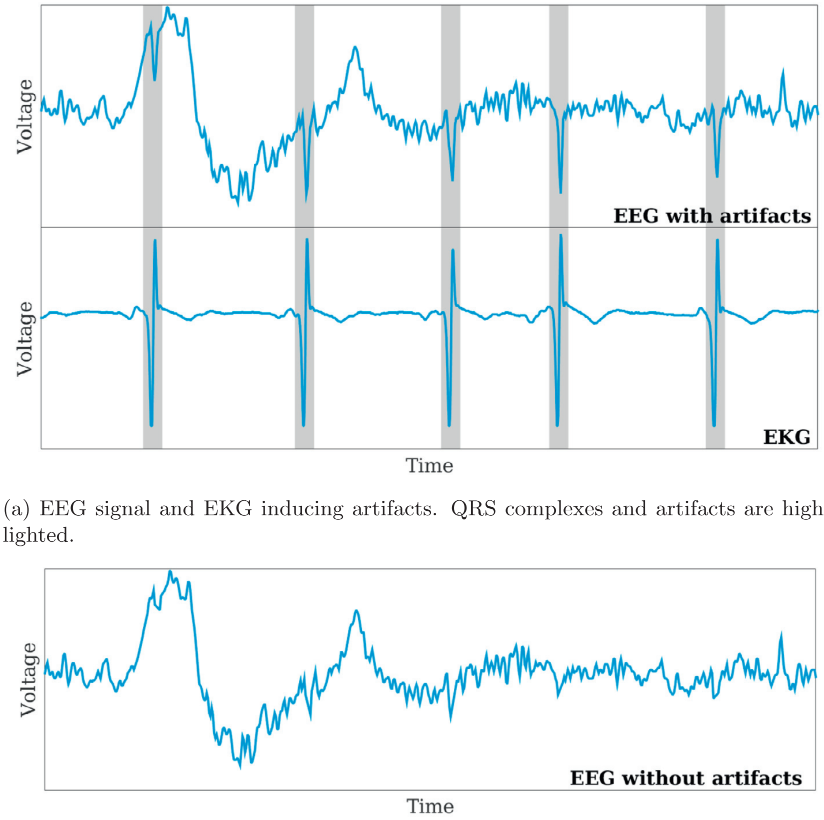
The three selected signals are processed using different techniques. After removing the artifacts, both EEG signals are studied in the frequency domain whereas the EMG signal is explored in the time domain. In both cases, a sliding window is used to select the relevant intervals employed in further steps.

2.1.1. Signal conditioning

The EEG signal and other related biosignals usually present artifacts that can mislead their interpretation [14]. The most common artifact, in the data used in this work, is the one induced by the EKG: a peak in the EEG at the same time as a QRS complex in the EKG. To remove these artifacts, the beginning and the end of the QRS complexes are located over the EKG signal. To locate these points, we studied the amplitude of the first derivative of the EKG, selecting the highest values. As not every complex causes an artifact, the original EEG is only corrected in those segments where an artifact is detected. To correct these segments, we interpolate the signal between the start and the end of the artifact. Fig. 2 illustrates the conditioning process.

2.1.2. Signal windowing

After both EEG signals have been conditioned, a windowing process takes place. The three available signals, the two EEG derivations, and the submental EMG, are studied using a sliding window.



(a) EEG signal and EKG inducing artifacts. QRS complexes and artifacts are highlighted.

(b) EEG signal after signal conditioning.

Fig. 2. Signal conditioning process to remove the EKG artifacts from the EEG.

The window used in the EEG derivations has a duration of 3 s, with a shifting step empirically selected. Theoretically, the shifting step should tend to 0, but we can improve the computational performance using higher values. According to the AASM [1], to score an EEG arousal, there must be an abrupt change in the alpha ($\alpha = 8$–12 Hz), theta ($\theta = 4$–7 Hz) and/or frequencies greater than 16 Hz. Transforming each window into the frequency domain, using a Hamming function and the Fourier transform, the power of the window is obtained following the formula:

$$power = \frac{1}{n^2}\sum_{i=1}^{n}|X(n)|^2$$

where n is the number of samples in the window and X the signal through the bandpass. A bandpass filter was applied for each frequency in the whole signal.

For each of the previously named frequencies (α, θ, and $>$ 16 Hz), a baseline over the average of the previous 10 s is created. Intervals with abrupt changes are selected when the ratio between the power and the baseline is greater than a threshold value selected empirically. From these different sets of intervals, one set for each band, those lasting less than 3 s (the minimum EEG arousal duration) are discarded. Moreover, two consecutive intervals from the same set, that last t_1 and t_2 with a non-overlapping time between them t_3, if $t_1 + t_2 > t_3$, $t_1 < t_3$ and $t_2 < t_3$, are replaced by one longer interval that includes both and lasts $t_1 + t_2 + t_3$. The window length and the baseline duration were chosen based on the definition of EEG arousal from the AASM. Fig. 3 shows the application of the windowing technique in one EEG derivation.

The process carried out in the EMG signal is similar to that described for the EEG derivations. Maintaining a 3 s window, we also selected an empirical value for the shifting step. To find the EMG activity related to an EEG arousal, the peak to peak amplitude in the window is studied. A baseline over the average of the previous 30 s is created. Again, each value over the baseline reflects an amplitude change. The selection of high activity intervals is also carried out when the ratio between both values (the amplitude value and the baseline) is greater than an empirically selected threshold. Intervals lasting less than 3 s are discarded. In this case, the window duration and the baseline length were selected according to our experimentation.

2.1.3. *Intervals grouping*

In previous studies, a 30 s based segmentation, called epoch, is used. We used these divisions to group the intervals of the different signals. The goal is to identify those intervals from the three signals occurring within the same epoch. With all those intervals occurring in the same epoch, using the interval middle point to select the epoch an interval belongs to, we form a group choosing one interval from each signal. If there is no interval for one signal, no group is formed, whereas if there is more than one, we select the one with the highest power on the frequency bands mentioned in Section 2.1.2. For example, as we see in Fig. 4, in epoch i there is only one interval for each signal, obtaining the group in a straightforward manner. In epoch $i + 1$ there is no interval in one of the

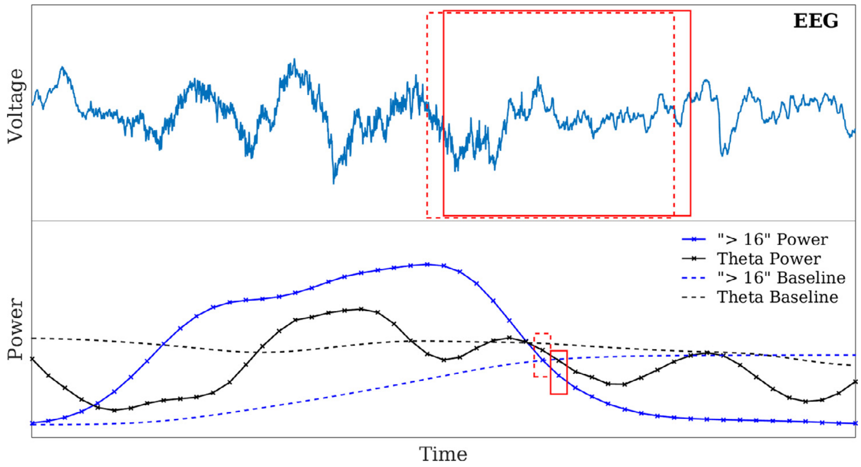


Fig. 3. Windowing technique through the EEG derivations. Only θ and > 16 Hz bands are represented for clarity. Straight and dotted squares represent two consecutive windows and the power values obtained for these bands.

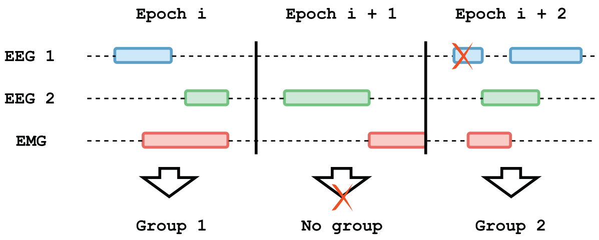


Fig. 4. Grouping intervals occurring in the same epoch.

EEG derivations, so no group is obtained. Finally, in epoch $i+2$, there are two intervals in the first EEG derivation, so to create the group, the one with the highest power value in α, θ, and $>$ 16 Hz bands is selected.

2.2. Machine learning models

From the groups obtained using the method described, a set of features is extracted and used over several classifications models. In this problem we have two possible classes, i.e. arousal and non-arousal. The best models, according to our experimentation, are selected to build combined approaches that improve the individual models' performance. These approaches are compared to ensemble methods to assess their performance.

2.2.1. Feature extraction

The classical problem of pattern classification can be represented by the relation $\langle v, d\rangle$ where v is a vector of features, and d the label indicating the classification. Applied to our problem, in general, $v = \{v_1, t_1, v_2, t_2, v_3, t_3, c\}$, where $v_1 = \{v_{1_1}, \ldots, v_{1_n}\}$ is the vector of features extracted from the interval of the first derivation of the EEG and t_1 the time at which the interval occurs; $v_2 = \{v_{2_1}, \ldots, v_{2_n}\}$ is the vector of features extracted from the interval of the second derivation of the EEG and t_2 the time at which the interval occurs; $v_3 = \{v_{3_1}, \ldots, v_{3_m}\}$ is the vector of features extracted from the interval of the EMG and t_3 the time at which the interval occurs; and $c = \{c_1, c_2\}$ is the vector of contextual features. As already mentioned, the use of the epoch division simplifies the problem because $t_1 = t_2 = t_3$ and thus, the vector of features stands as $v = \{v_1, v_2, v_3, c\}$.

Table 1
Intervals features description.

Signal	Feature	Description
EEG C3/A2	$v_{1_{1-5}}$	Total power on the band δ, θ, α, σ and $>$ 16 Hz
	$v_{1_{6-10}}$	Max. power on the band δ, θ, α, σ and $>$ 16 Hz
	$v_{1_{11-15}}$	Min. power on the band δ, θ, α, σ and $>$ 16 Hz
	$v_{1_{16-18}}$	Activity, mobility and complexity
	$v_{1_{19}}$	Duration
EEG C4/A1	$v_{2_{1-5}}$	Total power on the band δ, θ, α, σ and $>$ 16 Hz
	$v_{2_{6-10}}$	Max. power on the band δ, θ, α, σ and $>$ 16 Hz
	$v_{2_{11-15}}$	Min. power on the band δ, θ, α, σ and $>$ 16 Hz
	$v_{2_{16-18}}$	Activity, mobility and complexity
	$v_{2_{19}}$	Duration
EMG	v_{3_1}	Total amplitude
	v_{3_2}	Max. amplitude
	v_{3_3}	Min. amplitude
	v_{3_4}	Duration
Contextual	c_1	Sleep Stage
	c_2	Common time between the EEG intervals

Table 1 describes the complete set of features, v, that are extracted from each group of intervals. Regarding the EEG intervals, information of the power of the different bands is included. In Section 2.1.2, we have already described how to obtain the α, θ, and $>$ 16 Hz power values. These bands are completed adding the power of delta ($\delta = 0.5 - 4$ Hz) and sigma ($\sigma = 12$–15 Hz) bands. Lastly, Hjorth parameters are included, as it has been demonstrated that they are a good characterization of the EEG [15]. These parameters are defined as follows:

$$Activity = var(X(n))$$

$$Mobility = \sqrt{\frac{Activity(X'(n))}{Activity(X(n))}}$$

$$Complexity = \frac{Mobility(X'(n))}{Mobility(X(n))}$$

where X is the signal and X' the first derivative.

Regarding the EMG interval, we include information about the amplitude of the signal, obtaining the values as described in Section 2.1.2.

Finally, we include two contextual features: the sleep stage, automatically obtained following the method described in [16], as the detection of EEG arousals should be adapted as a function of the sleep stage (i.e., scoring of arousal during REM requires a concurrent increase in submental EMG); and the common time during which both EEG intervals appear simultaneously, if they do not overlap $c_2 = 0$, otherwise, $c_2 = min(end_{EEG1}, end_{EEG2}) - max(start_{EEG1}, start_{EEG2})$.

2.2.2. Classification models

As explained above, six classification models were considered for the EEG arousal detection task: Fisher's Linear Discriminant (FLD), Support Vector Machine (SVM), Artificial Neural Networks (ANN), Classification Trees (CT), k-Nearest Neighbor (k-NN) and Naive Bayes (NB).

2.2.3. Combined approach

The aforementioned classification models were compared to a mixed approach that combines the classification algorithms in two different manners: (1) a model following Shortliffe and Buchanan's certainty factors model, and (2) a linear combination approach.

The model proposed by Shortliffe and Buchanan (S&B) [17] is based on the definition of certainty factors (*CF*). Given the hypothesis, i.e., the presence of arousal, the *CF* can obtain a value between (−1, 1) where 1 completely asserts the hypothesis and −1 completely denies it. If for every classification, we translate the output of the model *i* to CF_i, the outputs of two models are combined as follows:

$$CF_{ij} = \begin{cases} CF_i + CF_j - CF_i \times CF_j & \text{if } CF_i > 0, CF_j > 0 \\ CF_i + CF_j + CF_i \times CF_j & \text{if } CF_i < 0, CF_j < 0 \\ \dfrac{CF_i + CF_j}{1 - min(|CF_i|, |CF_j|)} & \text{if } CF_i \times CF_j < 0 \end{cases}$$

The second approach proposed in this work is based on a linear combination. Maintaining the translation of the output to *CF*, we can define a linear combination of two individual models as follows:

$$CF_{ij} = w_i \times CF_i + w_j \times CF_j$$

where $w_i, w_j \in \mathbb{R}$ and $w_i + w_j = 1$. For n classification models, this can be generalized as:

$$CF = \sum_{i=1}^{n} w_i \times CF_i$$

where w_i is a weight factor verifying that $\sum_{i=1}^{n} w_i = 1, w_i \in \mathbb{R}$.

For both combined approaches, the obtained output must be used to decide whether the input is classified as EEG arousal or non-arousal. In this sense, the easiest solution is to use a threshold, classifying values greater than the threshold as arousals, and values lower than it as non-arousals.

2.2.4. Ensemble methods

An ensemble method (ensemble of classifiers) is a set of classifiers whose individual decisions are combined in some way to classify new examples. They usually outperform the accuracy reached by the considered models. A necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is if the classifiers are accurate and diverse [18]. An accurate classifier is one that has a classification error better than random guessing on new x values. Two classifiers are diverse if they make different errors on new data points. In the literature, ensemble methods is the name given to those combination of classifiers using only one classification model.

In this work, we compare our combined approaches against two ensemble methods: Random Forest (RF) and k-Nearest Neighbor Ensemble (k-NNE).

2.3. Performance measures

The performance of the method is evaluated in terms of the following measures:

- *The classification error* computed as the proportion of incorrectly classified positive and negative instances.
- *The sensitivity*, which quantifies the ability to identify positive instances correctly. It is the proportion of true positives that are correctly identified.
- *The specificity*, which quantifies the ability to identify negative instances correctly. It is the proportion of the true negatives that are correctly identified.
- *The AUC*, which compares the sensitivity and specificity simultaneously. In a two class problem, this is the average between both values.

3. Experimental procedures

The experiments conducted in this work use different data sets containing PSG recordings from real patients. All recordings were taken from the Sleep Heart Health Study (SHHS) [19], a database granted by the Case Western Reserve University, emerged from a multicenter cohort study implemented by the National Heart Lung and Blood Institute to determine the cardiovascular and other consequences of sleep-disordered breathing.

Each recording includes the annotations of different events, including EEG arousals, marked by the analysis of experts following the rules of the ASDA (current AASM) [20]. All the recordings were blind scored and anonymized. The montage includes the four signals used in this work, sampled at 125 Hz, right and left electrooculograms, thoracic and abdominal excursions, airflow, pulse oximetry, body position and ambient light.

From this study we built two different datasets:

- A first dataset containing 20 recordings. This dataset contains 2981 EEG arousal events in 23, 972 epochs. It was divided into two smaller subsets: 15 recordings for a training and validation set (T_R) containing 2202 arousals in 18, 094 epochs, and five recordings for a test set (T_S) containing 779 arousals in 5878 epochs. This dataset is used with two purposes. First, with the T_R we configured the machine learning models, both the individual models and the ensemble models. Second, with the classification over the T_S we compare the individual models, to select the best ones for the construction of the combined approaches.
- A second dataset containing 26 recordings, used previously in [21]. This dataset contains 4860 EEG arousals in 31,070 epochs. With this dataset we probed the combined approaches proposed, in a completely independent dataset. With this dataset we also compare the combined approaches against the individual and the ensemble models.

To select the empirical values mentioned in Section 2.1.2, we calculated the maximum sensitivity and the minimum specificity our method can achieve. If there is an EEG arousal in certain epoch, we can only score it if, previously, we found relevant intervals for both EEG derivations in that epoch. In other words, we can only score an EEG arousal in an epoch if we formed a valid group in that epoch. We define the maximum sensitivity using the number of epochs containing an EEG arousal and the number of those epochs where we found a valid group of events, and the minimum specificity with the number of epochs that do not contain an EEG arousal and the number of those epochs where we did not find a valid group. With these two parameters we selected the following values that are common for all the studied subjects:

- 0.2 s for the skipped time between windows in the EEG analysis.
- 1.5 times greater power than the baseline on power values for the threshold to select relevant EEG intervals.
- 0.4 s for the skipped time between windows in the EMG.
- 2 times greater amplitude than the baseline on amplitude values to select relevant EMG intervals.

Before any further experimentation with the classification models, some decisions regarding their configuration were made:

- *SVM*: an RBF kernel was selected because it maps the samples into a higher dimensional space, being able to handle the case when the relation between class labels and attributes is nonlinear.
- *ANN*: a feedforward network where one hidden layer was selected, trained with the conjugate gradient backpropagation algorithm [22].
- *k-NN*: the Spearman's Distance [23] to measure the distance between two elements was selected.
- *NB*: a kernel distribution [24] was selected, assuming the data does not follow the normal distribution.

As already explained, the T_R set was employed to configure the relevant parameters of all the models used in this work, using a 10-fold cross validation. It is important to notice that every dataset in this problem is highly unbalanced, with many more samples of non-arousal epochs than otherwise. This situation could easily lead to a bias classifier. To avoid this difficulty, we applied an under-sampling technique in the T_R set, preserving those epochs with presence of an EEG arousal and randomly selecting the same number of epochs without it. The test set remains unbalanced, as would be the case in a real life scenario in the presence of new, non-previously seen, PSG recordings. Those models achieving the lowest classification error with the 10-fold cross validation were the ones selected, after probing the following parameters:

- SVM: We attempted exponentially growing sequences for the parameter C $(2^{-11}, 2^{-9}, \dots, 2^{11})$ and the parameter S $(2^{-11}, 2^{-9}, \dots, 2^{11})$. Between the best values, a detailed grain search was conducted, choosing the final configuration as $C = 2^{11}$ and $S = 2^{-3}$.
- ANN: A similar approach was followed for the ANN varying the number of hidden neurons, H, from 2 to 2^8 in powers of two. The final value chosen was $H = 40$.
- CT: Different prune levels from 1 to 20 were tried. The final prune value selected was 15.
- k-NN: We tried the number of neighbors from 2 to 15, finally choosing 5 neighbors.

The ensemble methods were also configured using the same procedure, and selecting those models achieving the lowest classification error:

- RF: We tried different numbers of CT used to build the ensemble, finally selecting the value of 100.
- k-NNE: We tried combinations with different number of neighbors for the individual k-NN and length of subsets. The best combination found was 4 neighbors and 20 features in each subset.

Once we had configured all the models, we trained them with the complete T_R to conduct the remaining experiments.

The goal of the experiments carried out with the T_S is to specifically demonstrate the importance of two of the features included in the input space: the Sleep Stage and Hjorth's parameters. While the Sleep Stage is not usually included in EEG arousals studies, Hjorth's parameters are not considered good indicators for arousals [25]. Four feature selections experiments were conducted to meet the aforementioned goal: including all the features; excluding Hjorth's parameters; excluding the Sleep Stage; and excluding both features. . With these experiments, we not only validate our features but also establish the importance of each one. Moreover, these experiments allowed us to compare the proposed classification models.

To build our combined approaches, the best models in terms of classification error of these experiments were selected. These combined approaches also need a proper configuration to achieve the best performance. In the case of the S&B Combination, it has to be decided how to translate the classifiers output to a *CF* value, whereas for the Linear Combination the weights must be carefully chosen.

In order to obtain *CF* values from the outputs of the individual models, we followed different strategies for each kind of classifier. For the probabilistic ones where the output is the probability of the input representing an EEG arousal, the translation is straightforward, scaling its output values into the interval $(-1, 1)$. For the categorical classifiers in which the output is the class the input belongs to, we need a value to represent the class arousal and another one to represent the class non-arousal. From the experiments performed and to establish a similarity between the probabilistic and categorical classifiers, a value of 0.7 is used to represent the arousal class and a value of −0.7 to represent the non-arousal class. This assignment is based in the observation that the probabilistic classifiers do not tend to use high values (close to 1) to classify an input as arousal; in fact, values higher than 0.7 mean that the classifier is *almost sure* that the input represents an EEG arousal.

Regarding the linear combination model, the set of weights must be carefully chosen. We tried five different configurations:

1. $w_i = 1/n$
2. $w_i = error_i / \sum_{j=1}^{n} error_j$
3. $w_i = sens_i / \sum_{j=1}^{n} sens_j$
4. $w_i = spec_i / \sum_{j=1}^{n} spec_j$
5. $w_i = AUC_i / \sum_{j=1}^{n} AUC_j$

where n is the number of models selected for the combined approaches, and the performance measures were obtained with the experimentation done with the T_S set. The configuration chosen is the first one.

Finally, for both combinations, the decision between arousal or non-arousal for each combination output is made with a threshold. If the combination yields a value greater than the threshold, we classify as arousal and, otherwise, as non-arousal. These thresholds can obtain values in the interval $(-1, 1)$, which is the interval for the outputs of the combination methods. We selected those that achieve a similar AUC value compared to the individual classifiers, with the goal of minimizing the classification error. The final values for the S&B Combination and the Linear Combination were, respectively, 0.7 and 0.3.

4. Experimental results

Table 2 represents the four experiments conducted over the T_S set, including a trivial classifier that always decides there is no presence of EEG arousal. For those models that present variations over executions, because they used randomly initialized parameters, the results shown are the average of 15 executions.

First of all, the inclusion of the trivial classifier demonstrates that the comparison between individual classifiers cannot be made only with the classification error, but that in addition AUC and the sensitivity should be analyzed. Thus, classifier *A* is better than classifier *B* if the AUC achieved by *A* is higher than the AUC achieved by *B* and, at the same time, the classification error of *A* is lower than the classification error of *B*.

Table 2
Results using the T_S set from the individual classifiers showing the input features used. For those models varying with each execution, results shown are the average of 15 executions.

		Error	Sensitivity	Specificity	AUC
Trivial classifier		0.133	0	1	0.5
All the features	FLD	0.196	0.762	0.810	0.786
	SVM	0.134	0.815	0.874	0.845
	ANN	0.159	0.868	0.836	0.852
	CT	0.139	0.810	0.869	0.839
	k-NN	0.145	0.799	0.864	0.832
	NB	0.368	0.885	0.594	0.734
Without Hjorth parameters	FLD	0.171	0.737	0.843	0.790
	SVM	0.161	0.814	0.843	0.828
	ANN	0.213	0.876	0.774	0.825
	CT	0.142	0.796	0.868	0.832
	k-NN	0.149	0.798	0.859	0.828
	NB	0.347	0.873	0.620	0.746
Without Sleep Stage	FLD	0.200	0.745	0.808	0.777
	SVM	0.163	0.811	0.841	0.826
	ANN	0.207	0.862	0.782	0.822
	CT	0.155	0.789	0.853	0.821
	k-NN	0.169	0.794	0.836	0.815
	NB	0.367	0.878	0.596	0.737
With neither Hjorth nor Sleep Stage	FLD	0.171	0.721	0.845	0.783
	SVM	0.190	0.840	0.806	0.823
	ANN	0.191	0.840	0.804	0.822
	CT	0.163	0.810	0.841	0.825
	k-NN	0.176	0.798	0.828	0.813
	NB	0.345	0.872	0.623	0.747

It is clear that the inclusion of both features, Hjorth's parameters and Sleep Stage, improves the performance of the classifiers, as, in general, the methods obtain higher AUC values and lower classification errors. Results obtained adding only one of these two features are similar no matter which one used, and in both cases are worse than the results achieved with the complete set of features.

It is very noticeable that four of the models (SVM, ANN, CT and k-NN) outperform the other two (FLD and NB). Moreover, these latter two methods do not achieve better results, not even with the complete set of features. Thus, both FLD and NB classifiers will be excluded in future experiments and discussions.

Among the remaining classifiers, the experiment's performance varies. Focusing on the experiment with the complete set of features, the classification error achieved by the SVM and the CT is clearly lower than the error by the rest of the classifiers, while the AUC value improves with the ANN. The ANN also obtains the higher sensitivity, but as the classification error is also higher, it means that the number of false positives is also high. In this scenario, the best classifier is the SVM, with the second largest AUC value but the lowest classification error.

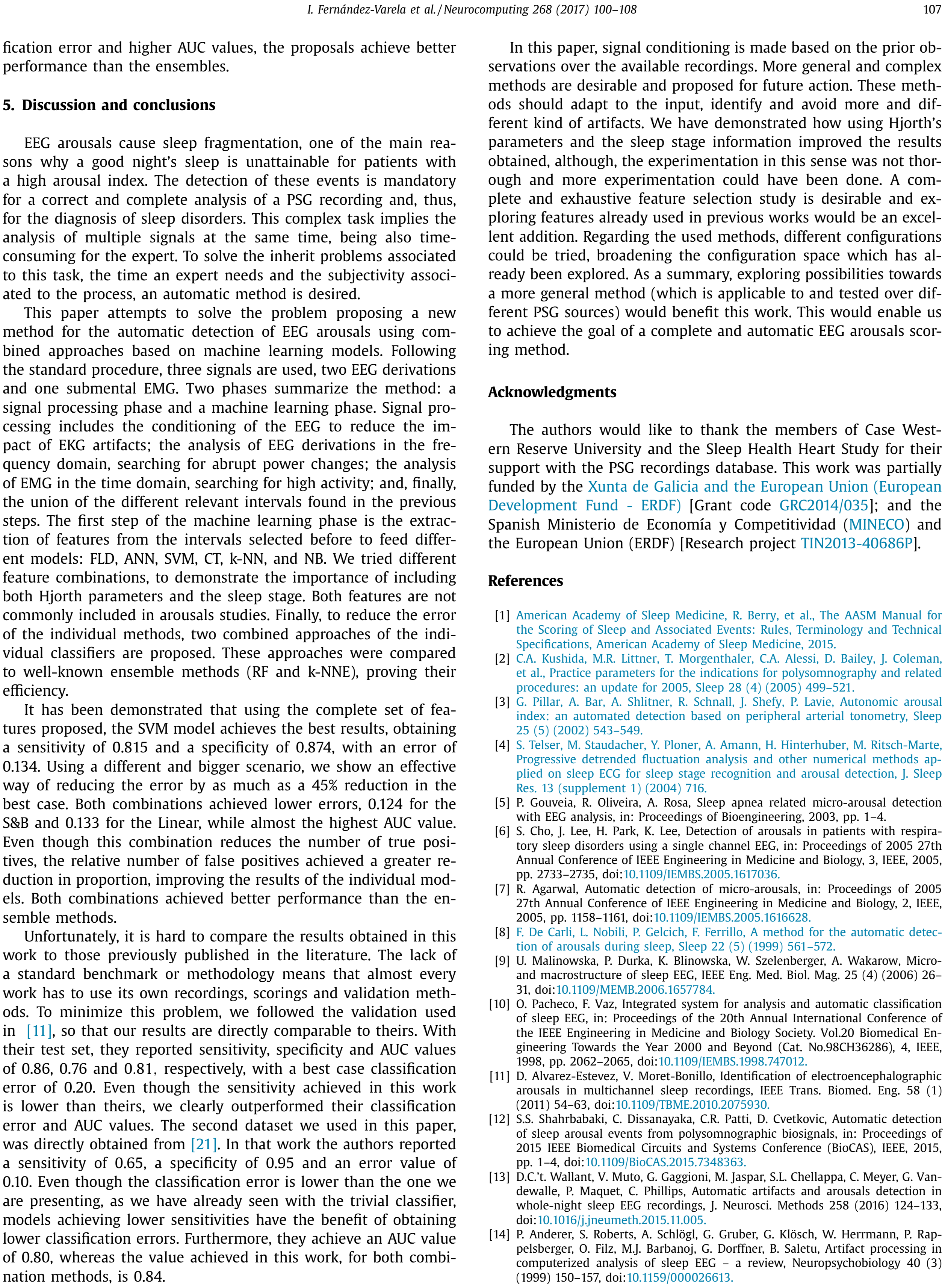
After comparing the individual models, we selected the four which achieved the best performance in terms of classification error and AUC values (SVM, ANN, CT, and k-NN), as the remaining classifiers are not trustworthy to use in the combined approaches. As explained in Section 3, the performance of these approaches was tested using a second dataset that contains 26 PSG recordings. This dataset was also used to test the individual models. This way we can compare them to the combined approaches and the ensemble models, and check the generalization capabilities of the method. The performance measures obtained using the different models are shown in Table 3.

Combining the different models we should expect a reduction in the error because the score of an arousal is more demanding, i.e., more models should agree than disagree. It is evident how the obtained results back up this theory. Both combinations clearly reduce the error, between a 16% in the worst case (the SVM compared against the Linear Combination) and a 45% in the best case (the ANN compared against the S&B Combination). Moreover, the higher AUC value is obtained with the Linear Combination, while the S&B combination achieved a higher AUC value than 3 of the individual classifiers but slightly lower than the SVM (by less than 1% lower). Of course, this improvement in classification error and AUC value does not come without a trade-off. In both combined approaches, sensitivity is lower than in most of the individual methods, as we have reduced the *lucky guesses*. The interesting fact is that this reduction in the scored arousals implies a higher decrease in the false positives, thus improving the specificity. As the decrease in false positives is higher than the loss of EEG arousals detected, the results obtained with the combinations are better than the ones obtained with the individual methods.

Table 3
Results from the combined approaches, the individual and the ensemble models using the dataset containing 26 PSG recordings.

	Error	Sensitivity	Specificity	AUC
Trivial classifier	0.156	0	1	0.5
SVM	0.159	0.845	0.840	0.843
ANN	0.224	0.900	0.754	0.827
CT	0.161	0.785	0.849	0.817
k-NN	0.173	0.814	0.829	0.822
S&B combination	0.124	0.781	0.893	0.837
Linear combination	0.133	0.810	0.878	0.844
RF	0.160	0.844	0.839	0.842
k-NNE	0.328	0.908	0.629	0.768

It is clear that both approaches outperform the ensemble methods. While the RF achieved slightly better results than the CT, the k-NNE obtained higher error and lower AUC values than the individual k-NN. The reason for this problem is that we could not select the same distance function because it did not apply to some of the feature subsets. Actually, the goal of including both ensembles was to validate and compare the performance of our combined approaches. As we see in the results, both in terms of lower classi-



fication error and higher AUC values, the proposals achieve better performance than the ensembles.

5. Discussion and conclusions

EEG arousals cause sleep fragmentation, one of the main reasons why a good night's sleep is unattainable for patients with a high arousal index. The detection of these events is mandatory for a correct and complete analysis of a PSG recording and, thus, for the diagnosis of sleep disorders. This complex task implies the analysis of multiple signals at the same time, being also time-consuming for the expert. To solve the inherit problems associated to this task, the time an expert needs and the subjectivity associated to the process, an automatic method is desired.

This paper attempts to solve the problem proposing a new method for the automatic detection of EEG arousals using combined approaches based on machine learning models. Following the standard procedure, three signals are used, two EEG derivations and one submental EMG. Two phases summarize the method: a signal processing phase and a machine learning phase. Signal processing includes the conditioning of the EEG to reduce the impact of EKG artifacts; the analysis of EEG derivations in the frequency domain, searching for abrupt power changes; the analysis of EMG in the time domain, searching for high activity; and, finally, the union of the different relevant intervals found in the previous steps. The first step of the machine learning phase is the extraction of features from the intervals selected before to feed different models: FLD, ANN, SVM, CT, k-NN, and NB. We tried different feature combinations, to demonstrate the importance of including both Hjorth parameters and the sleep stage. Both features are not commonly included in arousals studies. Finally, to reduce the error of the individual methods, two combined approaches of the individual classifiers are proposed. These approaches were compared to well-known ensemble methods (RF and k-NNE), proving their efficiency.

It has been demonstrated that using the complete set of features proposed, the SVM model achieves the best results, obtaining a sensitivity of 0.815 and a specificity of 0.874, with an error of 0.134. Using a different and bigger scenario, we show an effective way of reducing the error by as much as a 45% reduction in the best case. Both combinations achieved lower errors, 0.124 for the S&B and 0.133 for the Linear, while almost the highest AUC value. Even though this combination reduces the number of true positives, the relative number of false positives achieved a greater reduction in proportion, improving the results of the individual models. Both combinations achieved better performance than the ensemble methods.

Unfortunately, it is hard to compare the results obtained in this work to those previously published in the literature. The lack of a standard benchmark or methodology means that almost every work has to use its own recordings, scorings and validation methods. To minimize this problem, we followed the validation used in [11], so that our results are directly comparable to theirs. With their test set, they reported sensitivity, specificity and AUC values of 0.86, 0.76 and 0.81, respectively, with a best case classification error of 0.20. Even though the sensitivity achieved in this work is lower than theirs, we clearly outperformed their classification error and AUC values. The second dataset we used in this paper, was directly obtained from [21]. In that work the authors reported a sensitivity of 0.65, a specificity of 0.95 and an error value of 0.10. Even though the classification error is lower than the one we are presenting, as we have already seen with the trivial classifier, models achieving lower sensitivities have the benefit of obtaining lower classification errors. Furthermore, they achieve an AUC value of 0.80, whereas the value achieved in this work, for both combination methods, is 0.84.

In this paper, signal conditioning is made based on the prior observations over the available recordings. More general and complex methods are desirable and proposed for future action. These methods should adapt to the input, identify and avoid more and different kind of artifacts. We have demonstrated how using Hjorth's parameters and the sleep stage information improved the results obtained, although, the experimentation in this sense was not thorough and more experimentation could have been done. A complete and exhaustive feature selection study is desirable and exploring features already used in previous works would be an excellent addition. Regarding the used methods, different configurations could be tried, broadening the configuration space which has already been explored. As a summary, exploring possibilities towards a more general method (which is applicable to and tested over different PSG sources) would benefit this work. This would enable us to achieve the goal of a complete and automatic EEG arousals scoring method.

Acknowledgments

The authors would like to thank the members of Case Western Reserve University and the Sleep Health Heart Study for their support with the PSG recordings database. This work was partially funded by the Xunta de Galicia and the European Union (European Development Fund - ERDF) [Grant code GRC2014/035]; and the Spanish Ministerio de Economía y Competitividad (MINECO) and the European Union (ERDF) [Research project TIN2013-40686P].

References

[1] American Academy of Sleep Medicine, R. Berry, et al., The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications, American Academy of Sleep Medicine, 2015.

[2] C.A. Kushida, M.R. Littner, T. Morgenthaler, C.A. Alessi, D. Bailey, J. Coleman, et al., Practice parameters for the indications for polysomnography and related procedures: an update for 2005, Sleep 28 (4) (2005) 499–521.

[3] G. Pillar, A. Bar, A. Shlitner, R. Schnall, J. Shefy, P. Lavie, Autonomic arousal index: an automated detection based on peripheral arterial tonometry, Sleep 25 (5) (2002) 543–549.

[4] S. Telser, M. Staudacher, Y. Ploner, A. Amann, H. Hinterhuber, M. Ritsch-Marte, Progressive detrended fluctuation analysis and other numerical methods applied on sleep ECG for sleep stage recognition and arousal detection, J. Sleep Res. 13 (supplement 1) (2004) 716.

[5] P. Gouveia, R. Oliveira, A. Rosa, Sleep apnea related micro-arousal detection with EEG analysis, in: Proceedings of Bioengineering, 2003, pp. 1–4.

[6] S. Cho, J. Lee, H. Park, K. Lee, Detection of arousals in patients with respiratory sleep disorders using a single channel EEG, in: Proceedings of 2005 27th Annual Conference of IEEE Engineering in Medicine and Biology, 3, IEEE, 2005, pp. 2733–2735, doi:10.1109/IEMBS.2005.1617036.

[7] R. Agarwal, Automatic detection of micro-arousals, in: Proceedings of 2005 27th Annual Conference of IEEE Engineering in Medicine and Biology, 2, IEEE, 2005, pp. 1158–1161, doi:10.1109/IEMBS.2005.1616628.

[8] F. De Carli, L. Nobili, P. Gelcich, F. Ferrillo, A method for the automatic detection of arousals during sleep, Sleep 22 (5) (1999) 561–572.

[9] U. Malinowska, P. Durka, K. Blinowska, W. Szelenberger, A. Wakarow, Micro- and macrostructure of sleep EEG, IEEE Eng. Med. Biol. Mag. 25 (4) (2006) 26–31, doi:10.1109/MEMB.2006.1657784.

[10] O. Pacheco, F. Vaz, Integrated system for analysis and automatic classification of sleep EEG, in: Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol.20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No.98CH36286), 4, IEEE, 1998, pp. 2062–2065, doi:10.1109/IEMBS.1998.747012.

[11] D. Alvarez-Estevez, V. Moret-Bonillo, Identification of electroencephalographic arousals in multichannel sleep recordings, IEEE Trans. Biomed. Eng. 58 (1) (2011) 54–63, doi:10.1109/TBME.2010.2075930.

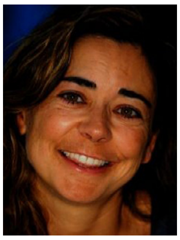
[12] S.S. Shahrbabaki, C. Dissanayaka, C.R. Patti, D. Cvetkovic, Automatic detection of sleep arousal events from polysomnographic biosignals, in: Proceedings of 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS), IEEE, 2015, pp. 1–4, doi:10.1109/BioCAS.2015.7348363.

[13] D.C.'t. Wallant, V. Muto, G. Gaggioni, M. Jaspar, S.L. Chellappa, C. Meyer, G. Vandewalle, P. Maquet, C. Phillips, Automatic artifacts and arousals detection in whole-night sleep EEG recordings, J. Neurosci. Methods 258 (2016) 124–133, doi:10.1016/j.jneumeth.2015.11.005.

[14] P. Anderer, S. Roberts, A. Schlögl, G. Gruber, G. Klösch, W. Herrmann, P. Rappelsberger, O. Filz, M.J. Barbanoj, G. Dorffner, B. Saletu, Artifact processing in computerized analysis of sleep EEG – a review, Neuropsychobiology 40 (3) (1999) 150–157, doi:10.1159/000026613.

[15] B. Hjorth, Eeg analysis based on time domain properties, Electroencephalogr. Clin. Neurophysiol. 29 (3) (1970) 306–310.
[16] D. Álvarez Estévez, J.M. Fernández-Pastoriza, E. Hernández-Pereira, V. Moret-Bonillo, A method for the automatic analysis of the sleep macrostructure in continuum, Expert Syst. Appl. 40 (5) (2013) 1796–1803.
[17] E.H. Shortliffe, B.G. Buchanan, A model of inexact reasoning in medicine, Math. Biosci. 23 (3) (1975) 351–379.
[18] L.K. Hansen, P. Salamon, Neural network ensembles, IEEE Trans. Pattern Anal. Mach. Intell. 12 (10) (1990) 993–1001.
[19] S.F. Quan, B.V. Howard, C. Iber, J.P. Kiley, F.J. Nieto, G.T. O'Connor, D.M. Rapoport, S. Redline, J. Robbins, J. Samet, et al., The sleep heart health study: design, rationale, and methods, Sleep 20 (1998) 1077–1085.
[20] E. Arousals, Scoring rules and examples. a preliminary report from the sleep disorders atlas task force of the American sleep disorders association, Sleep 15 (2) (1992) 173–184.
[21] D. Álvarez Estévez, Diagnosis of the sleep apnea-hypopnea syndrome. A comprehensive approach through an intelligent system to support medical decision, Universidade da Coruña, Departamento de Computación, 2012 (Ph.D. thesis).
[22] M.F. Møller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Netw. 6 (4) (1993) 525–533.
[23] C. Spearman, The proof and measurement of association between two things, Am. J. Psychol. 15 (1) (1904) 72–101.
[24] J.-M. Bernard, F. Ruggeri, A. Pérez, P. Larrañaga, I. Inza, Special section on the imprecise Dirichlet model and special section on Bayesian robustness (issues in imprecise probability) Bayesian classifiers based on kernel density estimation: flexible classifiers, Int. J. Approx. Reason. 50 (2) (2009) 341–362. http://dx.doi.org/10.1016/j.ijar.2008.08.008.
[25] M.J. Drinnan, A. Murray, J.E. White, A.J. Smithson, C.J. Griffiths, G.J. Gibson, Automated recognition of EEG changes accompanying arousal in respiratory sleep disorders, Sleep 19 (4) (1996) 296–303.

Isaac Fernández-Varela was born in Ferrol, Spain, in 1988. He graduated in computer science from the Universidade da Coruña, in 2012, and obtained a Master in High Performance Computing in 2013. He is currently working toward the Ph.D. in the application of artificial intelligent techniques in the field of sleep studies. His current research interests include machine learning and signal processing.



Elena Hernández-Pereira graduated in computer science from the Universidade da Coruña (Spain) in 1995. In 2000, she received her Ph.D. degree working in the area of the application of Artificial Intelligent techniques to sleep apnea diagnosis. She is currently an Associate Professor in the Computer Science Department, Universidade da Coruña. Her current research interests include machine learning, signal processing and medical decision support systems.

Diego Álvarez-Estévez was born in Ourense, Spain, in 1982. He graduated in computer science from the Universidade da Coruña, in 2007. He earned his Doctorate (cum Laude) in 2012, for his work on the application of artificial intelligence techniques to diagnose sleep apnea-hypopnea syndrome. In 2013 he joined the Sleep Center of Medisch Centrum Haaglanden in The Hague, Netherlands. His current research interests include computer-based sleep research, machine learning and biomedical signal processing.

Vicente Moret-Bonillo was born in Valencia, Spain, in 1962. He graduated with a degree in physical chemistry from the University of Santiago de Compostela, Spain, in 1984, and the first Postgraduate degree of research on control and monitoring of hemodynamic variables in 1985. Since joining the University's Department of Applied Physics, he earned his Doctorate (cum Laude) in 1998 for work on the application of artificial intelligence techniques to respiratory treatment of patients dependent on mechanical ventilation. From 1988 through 1990, he was a Postdoctoral Fellow in the Department of Biomedical Engineering Research, Medical College of Georgia, Augusta. He is currently a Professor Titular de Universidad in the Department of Computer Science, Universidade da Coruña, Spain. His main current research areas are knowledge representation, application of knowledge engineering techniques to dynamic systems, and performance analysis of intelligent systems. Dr. Moret-Bonillo is a memer of various scientific societies.

2.2 A simple and robust method for the automatic scoring of EEG arousals in polysomnographic recordings

- **Title:** A simple and robust method for the automatic scoring of EEG arousals in polysomnographic recordings
- **Authors:** Isaac Fernández-Varela, Diego Alvarez-Estevez, Elena Hernández-Pereira, Vicente Moret-Bonillo.
- **Journal:** Computers in Biology and Medicine.
- **Editorial:** Elsevier
- **ISSN:** 0010-4825
- **Year:** 2017
- **Volume:** 87
- **Pages:** 77-86
- **DOI:** 10.1016/J.COMPBIOMED.2017.05.011
- **Available in:** `https://www.sciencedirect.com/science/article/pii/S0010482517301336`

Computers in Biology and Medicine 87 (2017) 77–86

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

A simple and robust method for the automatic scoring of EEG arousals in polysomnographic recordings

Isaac Fernández-Varela [a], [*], Diego Alvarez-Estevez [b], Elena Hernández-Pereira [a], Vicente Moret-Bonillo [a]

[a] *Universidade da Coruña, Departamento de Computación, Facultade de Informática, Campus de Elviña s/n, 15071, A Coruña, Spain*
[b] *Sleep Center and Clinical Neurophysiology, Haaglanden Medisch Centrum, Lijnbaan 32, 2512 VA, The Hague, The Netherlands*

ARTICLE INFO

Keywords:
EEG arousals
Sleep studies
Signal processing
PSG recordings

ABSTRACT

Background: Clinical diagnosis of sleep disorders relies on the polysomnographic test to examine the neurophysiological markers of the sleep process. In this test, the recording of the electroencephalographic activity and the submental electromyogram is the source of the analysis for the detection of electroencephalographic arousals. The identification of these events is important for the evaluation of the sleep continuity because they cause the fragmentation of the normal sleep process. This work proposes a new technique for the automatic detection of arousals in polysomnographic recordings, presenting a non-computationally complex method with the idea of providing an easy integration with other algorithms.

Methods: The proposed algorithm combines different well-known signal analysis solutions to identify relevant arousal patterns with special emphasis on robustness and artifacts tolerance. It is a multistage method that after obtaining an initial set of events, improves the detection finding common EEG arousal patterns. Finally, false positives are discarded after examining each candidate within the context of clinical definitions.

Results: 22 polysomnographic recordings from real patients were used to validate the method. The results obtained were encouraging, achieving a precision value of 0.86 and a F_1 score value of 0.79. When compared with the gold standard, the method achieves a substantial agreement (Kappa coefficient of 0.78), with an almost perfect agreement with ten recordings.

Conclusions: The algorithm designed achieved encouraging results and shows robust behavior in presence of signal artifacts. Its low-coupled design allows its implementation on different development platforms, and an easy combination with other methods.

1. Introduction

Sleep disorders affect a major part of the population. Just as an example, between 30% and 40% of adults complain of insomnia, and between 5% and 15% of sleepiness [1]. Good sleep is essential for good health, and the consequences of bad sleep have been reported broadly [2]. Clinical diagnosis of sleep disorders relies nowadays on different procedures, however, the so-called polysomnographic test (PSG) still occupies a central role as the standard technique to examine the neurophysiological markers of the sleep process [3].

Among the different physiological data collected during the PSG, the recording of the electroencephalographic (EEG) activity and of the submental electromyogram (EMG) is the source of the analysis for the detection of transient events in the form of EEG arousals. These events are of interest in the context of evaluating a subject's sleep continuity: at the microstructural level of sleep, an EEG arousal represents an event triggering an awakening activity. A high presence of these events therefore provokes the fragmentation of the normal sleep process preventing restful sleep [4]. Specifically, according to the AASM [5], an EEG arousal is an abrupt shift in the EEG frequency including alpha, theta and/or frequencies greater than 16 Hz (but not spindles) that lasts at least 3 s, with at least 10 s of previous stable sleep. Also, during Rapid Eye Movement (REM) sleep stage, a concurrent increase in submental electromyography (EMG) is required.

Visual examination of the entire PSG for the scoring of these events is costly, due to the complexity of the analysis and the huge amount of data recorded per night. To help the clinician in the process, therefore, different works have explored several possibilities to automate the

[*] Corresponding author.
E-mail address: isaac.fvarela@udc.es (I. Fernández-Varela).

http://dx.doi.org/10.1016/j.compbiomed.2017.05.011
Received 9 March 2017; Received in revised form 10 May 2017; Accepted 11 May 2017

detection procedure.

Pacheco and Vaz [6] (1998), implemented a system that obtained frequency features from one central and one occipital EEG derivations and power features from the EMG. The obtained values were used to feed a K-means classifier, discarding false detected events with a context rule module. Zamora and Tarassenko [7] (1999), compared the use of autoregressive models and a bank of bandpass filters as a feature extraction technique, from one EEG derivation, to train and test a radial basis function neural network. De Carli et al. [8] (1999), followed a multichannel approach, analyzing two EEG derivations and extracting features by means of the wavelet transform. They also obtained transient increases in EMG muscle activity using a weighed moving average operator. Multichannel data was then integrated using a threshold to select probable arousals. Pillar et al. [9] (2002), proposed the use of peripheral arterial tonometry (PAT), studying the amplitude in combination with detected increases in pulse rate to obtain a derived PAT-arousal index. Cho et al. [10] (2005), studied the use of a single EEG derivation and proposed a method based on time-frequency characteristics, followed by classification using Support Vector Machines. Agarwal [11] (2005), extracted frequency features from the alpha and beta bands, from one occipital, and one central derivations respectively. Then, they applied segmentation and statistical methods over the features. Malinowska et al. [12] (2006), applied a matched pursuit procedure in one EEG derivation, and studied the deviation of the EMG amplitude during REM stages. Sugi et al. [13] (2009) reported a method in which they analyzed multichannel data, particularized for patients with sleep apnea syndrome. Four channels of EEG were used, extracting their periodogram features. Respiratory state (pressure and temperature) and chin and tibialis EMG characteristics were computed as well. Adaptive detection thresholds were developed on the basis of the respiratory and muscle activity for the detection of respiratory-related EEG arousals. The method included automatic detection of the sleep periods. Shmiel et al. [14] (2009) on their part, proposed the use of data-mining techniques for the extraction of arousal patterns implicit in the signals of EEG, EMG, pulse rate and arterial oxygen saturation. Alvarez-Estevez and Moret-Bonillo [15] (2011) developed a marker based on the spectral features of the alpha and beta bands, followed by a feature extraction process using two central EEG, and one submental EMG derivations. Different machine learning models were used to evaluate the relevant features to identify the arousal patterns. The application of feature selection methods to reduce input dimensionality was studied in Alvarez-Estevez et al. [16], and the method was more recently revised in Fernández-Varela et al. [17]. Behera et al. [18] (2014), also followed the aforementioned study [15], adding more features to the input of an artificial neural network. Finally, Wallant et al. [19] (2016), have recently proposed a method for the automatic detection of artifacts which included a module for the characterization of EEG arousals.

This work proposes a new method based on a multi-channel analysis context. The goal is to build a robust and efficient method, but simple at the same time, to automatically score EEG arousals and help the clinician during the PSG examination task. The method is designed to allow an easy integration with different platforms or applications. To achieve this end, well-known signal processing routines, easily implementable in any programming language are used. The resulting algorithm is configurable, and it can be executed using one EEG and one chin EMG derivations, while the presence of an extra EKG derivation is desirable when the aforementioned signals present artifacts caused by the interference of the heart beats.

2. Proposed method

Following the design shown in Fig. 1, we propose a multistage method that begins with signal conditioning and then obtains an initial set of events – candidate EEG arousals – based on frequency measures over the signals. Within this initial set of events, different methods search for the arousal pattern to improve the detection. Several features from each event are studied for this purpose. Finally, false positives are discarded, examining the resulting patterns within the context of clinical definitions.

In the following sections each stage of the method is described.

2.1. Signal conditioning

In this stage, every signal is Notch filtered to remove mains interference, which in our dataset happens at 50 Hz. Then, a high-pass filter (cut-off 15 Hz) is applied to the EMG signal to eliminate low frequencies not related to the chin muscle activity. Detailed information on the implementation of both filters can be found in Alvarez-Estevez [20].

The resulting EEG and EMG signals are used to build an adaptive filter that removes electrocardiogram (EKG) artifacts caused by the interference of the heart beats. For this purpose, first the EKG beat series is obtained using a standard QRS detection algorithm [21]. Then an EKG reliability analysis is performed to determine which intervals of the EKG series are reliable enough to be included in the adaptive filtering process. Both the EKG reliability analysis and the design of the adaptive EKG filter are described below.

2.1.1. EKG reliability analysis

The EKG signal is reliable if it has a low level of noise and is not excessively affected by artifacts. Our hypothesis is that within reliable signals, QRS peaks in the EKG would show similar amplitudes and would be regularly distributed. We have implemented an algorithm to analyze the reliability of the EKG signal using a 30 s window. The algorithm works over a EKG-derived contour signal, namely the lump-signal, where each lump corresponds to a QRS-complex occurrence. This signal is obtained using the filtering pipeline proposed in Hamilton [22]. We have modified this pipeline, adding a first Notch step, and replacing the low-pass and high-pass filtering concatenation by our own 8–16 Hz band-pass filtering implementation. Further, the output is amplitude normalized to avoid inter-individual amplitude differences. More details about the derivation of the lump-signal and of the filter's implementation have been published elsewhere [23].

Using the derived lump signal, we compare the value of the signal against a reference baseline to find the peaks. In this respect, we use a 10 s moving window without overlapping. For each window we use the average amplitude of the lump signal as the reference baseline. Peaks are defined as the maximum values of each region crossing over the baseline reference.

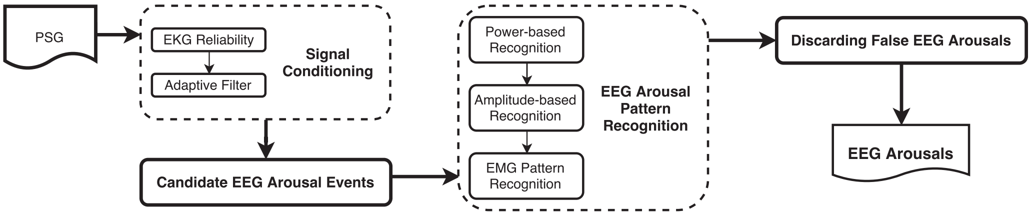

PSG
EKG Reliability
Adaptive Filter
Signal Conditioning
Candidate EEG Arousal Events
Power-based Recognition
Amplitude-based Recognition
EMG Pattern Recognition
EEG Arousal Pattern Recognition
Discarding False EEG Arousals
EEG Arousals

Fig. 1. Outline of the proposed method.

Finally, for each 30 s window, reliability is studied using the amplitude of the corresponding peaks. The following ratio is calculated: $r = mean(A)/std(A)$, where A is the window's peaks vector. We have empirically determined that the signal is reliable if $r \leq 0.05$. The interested reader is aimed to check Alvarez-Estevez et al. [23] for more details.

2.1.2. Adaptive EKG artifacts removal

The EKG artifact cancellation algorithm works as follows. For each (reliable) R-peak in the EKG beat series, we take a temporal 0.5 s window $W(t)$ in the corresponding signal (EEG or EMG), centered around the R-peak occurrence. The window $W(t)$ is updated with an adaptive filtering template T, using a memory factor α of 0.01:

$$T[t] = (1 - \alpha) \times T[t - 1] + \alpha \times W[t]$$

The resulting filtering template is then "tapered" using a Hann function, thus most of the filtering occurs around the peak occurrence, fading away toward the window extremes. Finally, the resulting artifact template is subtracted from the original signal. If the corresponding EKG interval has been classified as unreliable, $W(t)$ is set to the zero window ($W(t) = 0$) and therefore it does not contribute to the update of the artifact template. Neither the filtering is done at the corresponding time instant. Fig. 2 shows the effect of the resulting algorithm over a sample of the EEG signal with presence of EKG artifact.

2.2. Detection of candidate EEG arousal events

After the preprocessing of the input signals, we search for abrupt EEG frequency changes, that is, candidate arousal events. To do so, we compare the power content at a certain instant of time with the corresponding baseline levels from the immediate past instants.

Specifically, for the conditioned EEG signal, we evaluate the instant power using the Short-Time Fourier Transform (STFT) with a 3 s sliding window with Hamming transformation and a shifting step of 0.2 s. For each window $w(k)$, power content in the alpha (8–12 Hz) and in the beta (>16 Hz) bands is estimated by averaging the corresponding squared periodogram regions.

For each frequency band (alpha and beta), the baseline level of the current window is calculated by averaging the respective power values from the previous 10 s windows. Candidate arousal events are then marked between periods of relevant threshold crossing over the corresponding baseline values. Specifically, events in the alpha band (alpha events) are marked when instant power values cross 2.5 times over the alpha baseline, and events in the beta band (beta events) are marked when the corresponding power values cross over 2 times the respective beta baseline.

2.3. EEG arousal pattern recognition

Candidate arousal regions detected by the aforementioned process represent triggering events signaling the start of candidate EEG arousals. Once these events have been marked, we study them individually to recognize the actual arousal patterns with both EEG and EMG relevant features. Eventually, several candidate EEG arousals can be merged together or the initial arousal region can be adjusted, usually by extending the event's end, if recognizable arousal activity follows. For this purpose, a pattern matching approach is used which is described next. For clarity purposes, let us from now on assume that each candidate event is characterized by its respective onset and offset times t_1 and t_2. We will refer to the instant power values of the window being processed at the time instant t as $w(t)$.

2.3.1. EEG power-based pattern recognition

The first type of patterns we try to recognize are power related, and as such, we use the instant power values series $w(t)$ obtained by the procedure described in Section 2.2. Indeed, internal frequency content variability during the occurrence of an arousal event and the sliding window effect might lead to situations such as the one presented in Fig. 3a and b, where an initial event has been detected, even though the frequency shift is still visually noticeable after the initial offset time. Notice as well, that due to the effect of the increased (in this case beta) relative power content during the event, the concurrent associated baseline level increases as well, hampering a correct detection of the event's offset.

We can obtain more accurate event markers examining each candidate arousal and comparing the pre-event power activity to the future power values. A pre-event reference power value (V^2_{ref}) is calculated averaging the power series during the t_{ref} pre-event windows $[w(t_1 - t_{ref}), \ldots, w(t_1)]$, where $t_{ref} = 10$s for beta and 3 s for alpha power series. The event's future power values are then examined, comparing V^2_{ref} against V^2_f, which is defined as the mean power of the subsequent event's future 1 s windows: $V^2_f(k) = mean([w(t_2 + k - 1), \cdots, w(t_2 + k)])$, $k = 1, 2, \cdots$ If $V^2_f(k) > th \times V^2_{ref}$ the end of the event is updated, stopping when n consecutive windows fail meeting the previous condition. For the beta events we defined $th = 3$ and $n = 3$ and for the alpha events $th = 4$ and $n = 1$.

The event's end update is then validated, to ensure it is not a false detection, using two measures. First, either the mean power during the event or at least one of the future power values must be higher than th times V^2_{ref}, with $th = 5.5$ for beta and 6.5 for alpha events. Second, to avoid the false detection of artifacts in the form of amplitude peaks, the peak-to-peak amplitude of the EEG signal during the event is limited to be 8 times the peak-to-peak amplitude of the previous 10 s window. Fig. 3a and b shows the resulting delimitation of an EEG arousal before and after

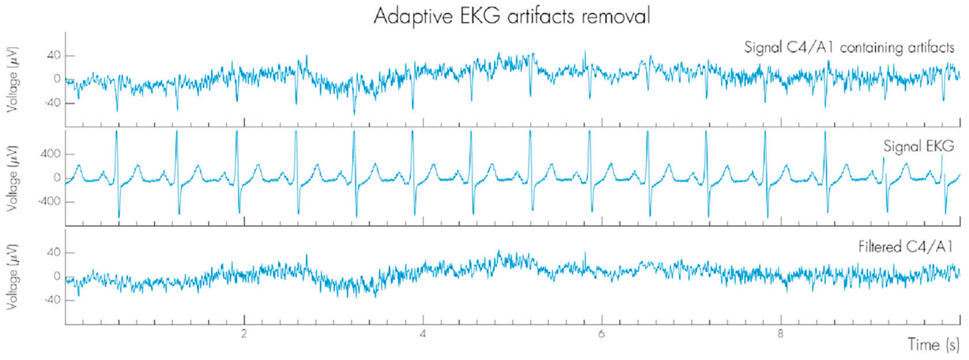


Fig. 2. Adaptive EKG artifact removal in the EEG signal.

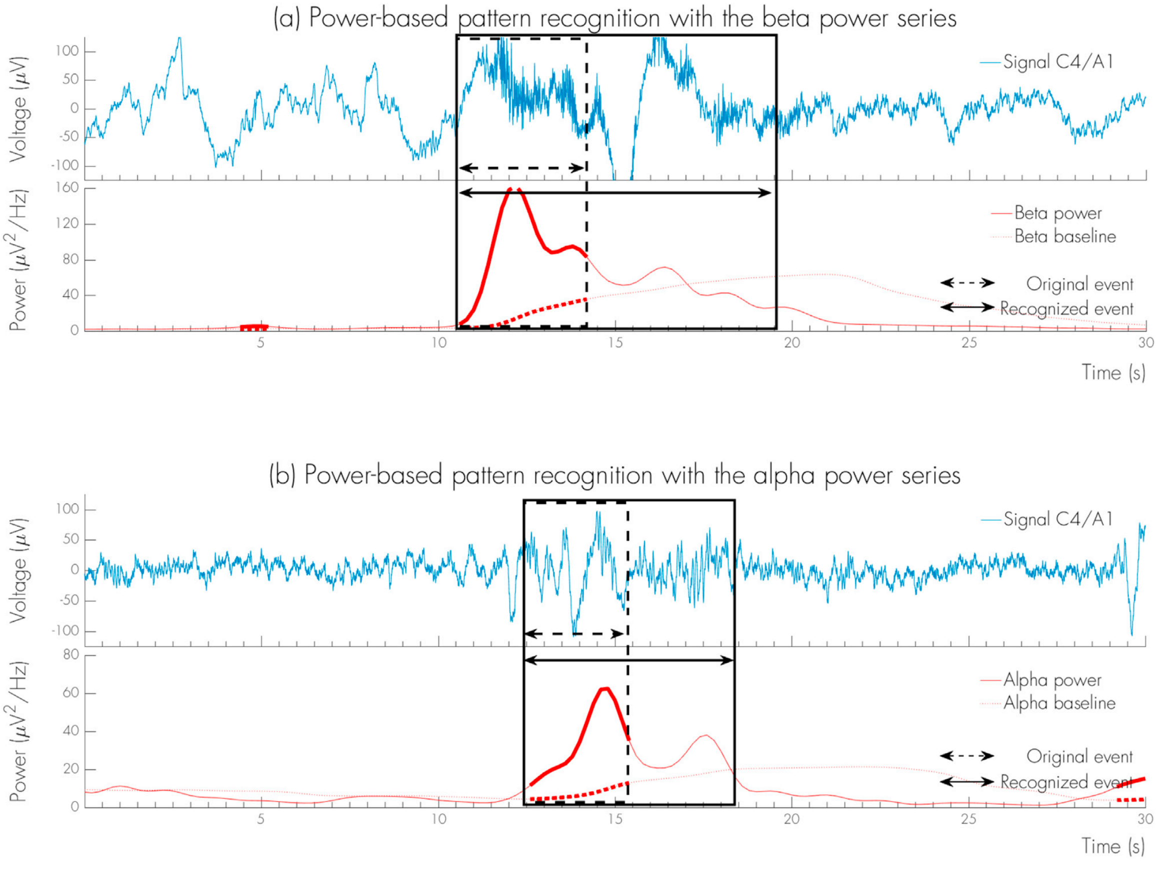


Fig. 3. Original event and power-based recognized pattern using the power series: (a) with the beta power series; and (b) with the alpha power series.

the execution of the above described procedure.

During this procedure, we have used different parameter settings for the beta and alpha power series. The different behavior of the corresponding power series justifies these variations. Besides, using the training data, the specific values on each case have been chosen to optimize the sensitivity of the detection while keeping the number of false positives low.

2.3.2. EEG amplitude-based pattern recognition

Even though the official AASM guidelines do not establish any prerequisite regarding the amplitude of the EEG, an EEG arousal usually involves a shift in the EEG signal amplitude. We use this particularity to improve the detection of EEG arousal patterns.

Hence, an amplitude change is detected when the ratio between the event's peak-to-peak amplitude and the corresponding average from its previous 5 1-s non-overlapping windows is greater than an empirically selected threshold, with a value of 4 both for alpha and beta events. If such an amplitude change is detected, then the subsequent 1 s window is appended to the analyzed event. The process is repeated with the new updated event, while the previous condition holds.

2.3.3. EMG pattern recognition

As arousals are usually accompanied by a concurrent EMG activation [5], to improve the EEG arousal pattern detection, we applied several methods based on this fact. These methods are described below.

EMG activity overlapping a candidate EEG event. Fig. 4a shows a typical situation in which chin EMG activity can be found overlapping the occurrence of an event detected in the EEG. In this situation we can improve the initial delimitation of the arousal event recognizing the active EMG region.

For this purpose, we compare the EMG amplitude during the original event (V_{ev}) against a reference value from the window surrounding the event (V_{ref}). The window surrounding the event goes from $t_1 - 15$ s to $t_2 +$ 15 s, but excluding the event. To obtain the reference value we use the mean peak-to-peak amplitude of the inner 0.1 s windows. If $V_{ev} > 1.4 \times V_{ref}$, relevant EMG activation is detected, and in such a case, the event is extended appending the consecutive 0.1 s window. The process is repeated while the condition holds.

Sometimes, though, the associated EMG activation might not be continuous, with short periods of lower amplitude alternating, which might mislead the previous method. This situation is shown in Fig. 4b.

To solve this problem, we compare the measures from different windows, using this time the rectified EMG. Relevant EMG activity is then recognized using a 3 s sliding window $w_{emg}(k)$, with a 0.5 s shifting step during the period $[t_1 - 3.5\text{ s}, t_2]$. The similitude between two windows (s_i) is defined as the mean difference between the i-th window ($w_{emg}(i)$) and the first window in the analysis period: $s_i = mean(w_{emg}(i) - w_{emg}(0))$. Using this measure, the regions in which $s_i > 2{*}s_1$, $1 < i \leq n$ are marked, and the following corresponding values are calculated $\mu_{diffs} = mean(s_i)$ and $sd_{diffs} = sd(s_i)$ $(1 < i \leq n)$. Finally the extension rule is applied, appending consecutive 0.5 s windows to the original EEG arousal event while $s_n > \mu_{diffs} - sd_{diffs}$.

Matching with non-overlapping EMG activity. EMG pattern detection can be improved as well accounting the fact that EMG activation might occur showing certain delay with respect to the initial delimited EEG event (see Fig. 4c).

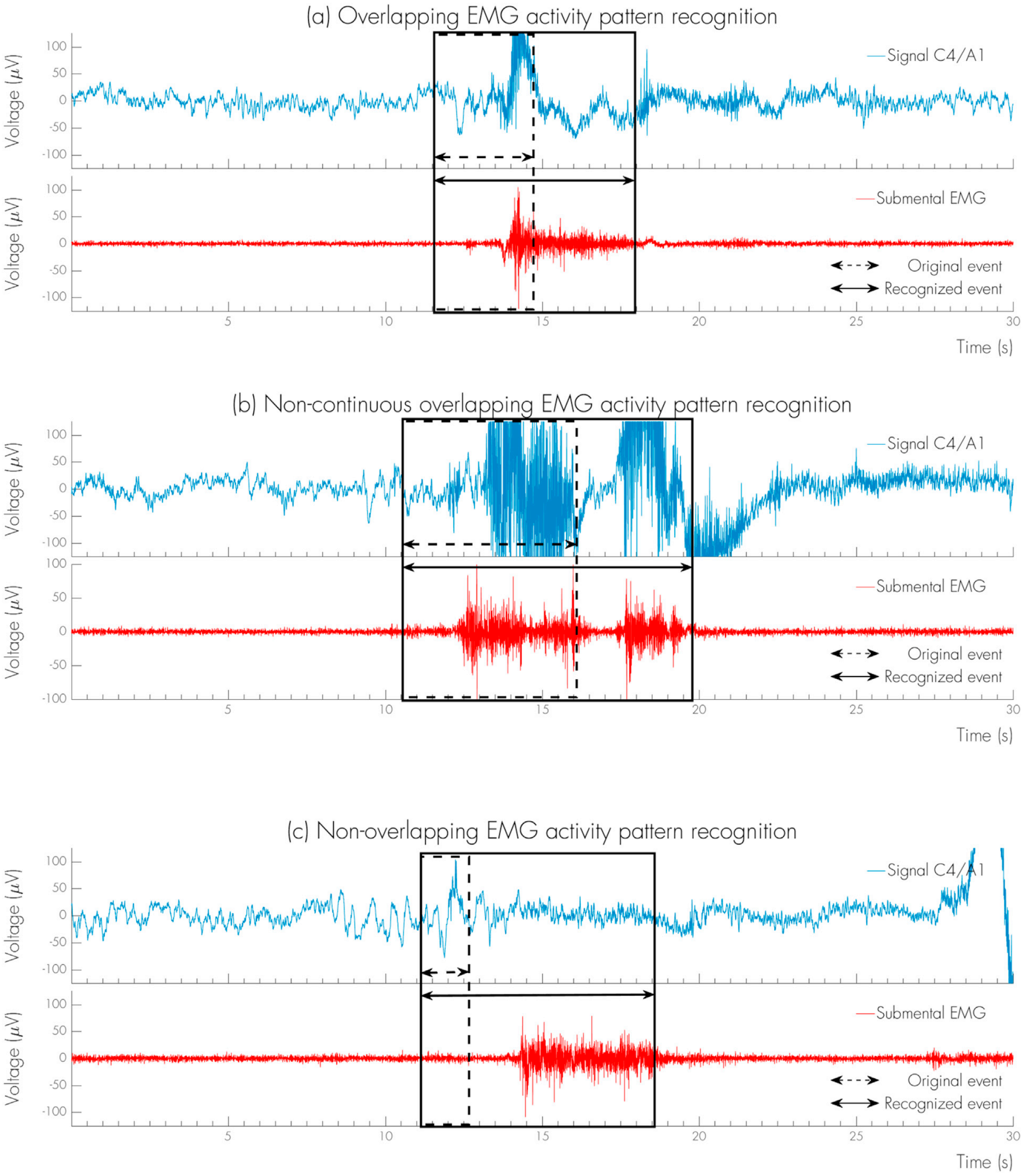


Fig. 4. Overview of the original and recognized event using the EMG signal with the method for each situation: (a) with overlapping EMG activity; (b)with non-continuous overlapping EMG activity; and (c) with non-overlapping EMG activity.

This situation is detected comparing the amplitude values from the future 0.5 s windows against a reference pre-event amplitude value, corresponding to the 10 s window before the event's start. These values are computed using the mean peak-to-peak amplitude of the corresponding 0.125 s inner windows. If two consecutive future windows have a value 2.5 times over the reference value, the event's end is updated to include the corresponding following windows. The process stops when the event is not updated after having analyzed the windows for the next 4 s.

2.4. Discarding false EEG arousals detections

To discard false positives and improve the overall detection, several post processing methods are applied. Each of these methods is inspired by the standard EEG arousal definition as found in the reference clinical guidelines [5].

Adjustment by event's duration. Events lasting less than 3 s are removed. Events lasting more than 15 s within the same epoch, are discarded as well, following clinical practice. Events overlapping two consecutive epochs though allowed, up to a full duration of 30 s. Furthermore, short duration alpha events (between 3 and 4.5 s) are also discarded as we have found these to be very subjective, and often cause a misdetection.

Detection of Sleep Spindles. Every detected arousal pattern is analyzed for possible presence of overlapping sleep spindle activity. Detection of sleep spindles is done using a band passed EEG signal between 1 and 32 Hz to avoid interference of higher frequencies. A 0.5 s sliding window – the minimum spindle duration – is used, crossing the whole arousal event with a 0.125 s shifting step. Whenever spindle activity is found (main frequency content of the window is in the spindle range 12–15 Hz) then the start of a spindle is marked, from which re-evaluation of the frequency content is performed by adding the successive 0.125 s windows. The spindle's end is marked when the main frequency content deviates out of the spindle range. The main frequency content associated to a window w is calculated as:

$$\mathrm{f}(w[n]) = \frac{\mathrm{c}(w[n])}{j-i}, i \leq n \leq j \tag{1}$$

where

$$\mathrm{c}(w[n]) = \left| \left\{ \mathrm{d}_1\left(\frac{|\mathrm{d}_1(w[n])|}{\mathrm{d}_1(w[n])} \right) > 0 \right\} \right|, \mathrm{d}_1(w[n]) = w\left[n\right] - w\left[n-1\right]$$

with $w[n]$ the sample of window w at instant n, i the first sample of w, and j the last sample of w.

After all the spindle periods are identified, the decision whereas the considered EEG arousal pattern is a false detection, or not, depends on the duration of the remaining spindle-free segments. If within the detected arousal pattern there is at least a period of 3 s or more, free of spindle activity, then the event is considered a true detection. Otherwise is discarded.

Discarding by absence of EMG activation during REM periods. During REM sleep periods a EEG arousal requires the presence of at least 1 s of concurrent EMG activation. We perform the detection of REM periods using the low muscle tone characteristic, under the assumption that REM sleep is the one with the lowest EMG amplitude [5]. The detection of these periods is done on a 30 s epoch basis, using the rectified EMG signal. A reference value is defined as the minimum peak-to-peak amplitude value of all the epochs. Epochs with a peak-to-peak amplitude lower than 2.5 times the reference value are associated to the low tone characteristic (i.e. they are estimated to belong to a REM period). Events during REM phases are evaluated comparing the peak-to-peak EMG amplitude during the event against the EMG amplitude of the epoch to which they belong. If the amplitude of the event is below 1.1 times the amplitude of the epoch, the event is not considered to show enough EMG activation. Thus, it is discarded.

Discard by absence of 10 s of previous stable sleep. To remove those events without at least 10 s of previous stable sleep, we follow two steps. First, for every event, if the next one follows with less than 10 s after the end of first, then the last one is removed. Second, those events happening in an epoch scored as W (awake) are removed. For this purpose, the algorithm needs as an input a classification of the epochs either as W or sleep, which is discussed in the following sections.

3. Results and validation

The patient database at the sleep center of the Haaglanden Medisch Centrum (The Netherlands) was used for the validation of our method. The study obtained the approval of the Medical Ethics Committee of the Southwest Holland region under reference METC 16-027. A set of 30 (16 males/14 females) in-lab PSG recordings with mean(SD) age of 50.4(20.8), Apnea Hypopnea Index (AHI) of 7.8(8.0), and Arousal Index (ArI) of 13.9(11.2) was randomly selected in order to match as close as possible the conditions of real common practice. Patients were referred to the sleep center in the context of different sleeping conditions of which from the previous dataset 16 presented a major Obstructive Sleep Apnea Syndrome (OSAS) component, and the rest were diagnosed respectively with complex OSAS (1), central SAS (2), hypersomnia and/or narcolepsy (2), REM Sleep Behavior Disorder (2), NREM parasomnia (1), and the rest (6) showed no evidence of major sleep disorders in the PSG, and therefore were considered as healthy subjects. No attempt was made to reject any recording due to the presence of signal artifacts, with the only condition that the quality of the recording was good enough for being scored by a human expert. Patient signals were acquired using SOMNOscreen^{TM} plus devices (SOMNOmedics, Germany) and digitized to the EDF+ format [24]. PSGs were afterwards analyzed offline by clinical experts in the course of common clinical practice. Scoring of events included, among others, the annotation of sleep stages and of EEG arousals. For the purposes of this study the results of the clinical scoring of EEG arousals were reviewed by an additional expert to achieve a consensus scoring. All the procedures were performed according to the last version of the AASM guidelines [5]. For the validation of our method one EEG derivation (C_4/A_1), the submental EMG, and the single-channel modified lead II EKG derivation were used. All signals were sampled at 256 Hz. Further the whole set was partitioned into two disjoint sets of 6 and 22 recordings respectively. The first one was used as training set to develop the method and to set the parameters. The second partition was used as independent validation set to assess the performance of the method.

The validation was made on a 30 s epoch basis. Every EEG arousal was assigned to a unique epoch, the one its middle point belongs to. Table 1 shows the validation results.

As it can be seen, the number of EEG arousals detected is slightly lower than the number of events scored by the experts (12% lower), as well as the number of false positives compared to the number of false negatives (48% lower). These situation was expected, as threshold settings were selected in the method with the idea of keeping the number of false positives relatively low. This settings are important to keep good precision values, as usually the proportion of recording time with presence of EEG arousal is relatively short, comparing it against the total recording time.

Indeed, it is worth noticing that in our dataset the classes distribution is highly unbalanced and dominated by the absence of event – only an 8% of the epochs contain an arousal event. Thus, in this domain, achieving a high specificity and a low error rate is not representative.

We can further demonstrate this point by taking into account the results shown in Table 2. In this table, the performance achieved by two trivial methods is calculated: the first one scores an event in every epoch, and the second one never scores an arousal.

The second trivial solution achieves an error of 0.078 and a specificity of 1, while in fact the method “never detects an EEG arousal”. Instead, metrics such as the F1-score and the Precision, which do not take into account the number of true negatives, or the Kappa index, which adjusts agreement due to chance, are better suited to study this problem. As it is shown in Table 1, our method achieves an average Kappa score of 0.775, which according to the interpretation scale made by Landis and Koch [25] is a substantial agreement. Kappa indices per recording range between 0.654 and 0.886, on 10 recordings reaching almost perfect agreement (kappa values over 0.80).

It is also interesting to compare how similar are the events scored by the expert to the ones scored by the proposed method, avoiding the established epoch granularity. For this purpose we can study the pairwise time differences between the positive matches (TP in Table 1). In Fig. 5 we compare each pair of events, showing the difference between their duration, and their onset and offset times. Positive values indicate that the expert's event is longer (or happens later in case of the onset/offset times) than the one scored by the automatic method. Both Anderson-Darling and Lilliefors statistical tests rejected the null normality hypothesis with $p < 0.001$ for all the three distributions. Their respective quartile values are shown in Table 3. The Wilcoxon signed rank test over the difference distributions showed that the respective medians were significantly different from zero in the case of the onset/offset deviations ($p < 0.001$), while for the event's duration the null hypothesis could not

Table 1
Epoch-based validation for the detection of EEG arousals. RN = Recording Number; TP = True Positives; FP = False Positives; TN = True Negatives; FN = False Negatives; Sens = Sensitivity; Spec = Specificity; Prec = Precision.

RN	# EEG Arousals											
	Expert	System	TP	FP	TN	FN	Error	Sens	Spec	Prec	F_1 score	Kappa
1	51	52	44	8	883	7	0.016	0.863	0.991	0.846	0.854	0.846
2	148	164	124	40	628	24	0.078	0.838	0.940	0.756	0.795	0.747
3	51	37	31	6	981	20	0.025	0.608	0.994	0.838	0.705	0.692
4	24	30	24	6	924	0	0.006	1.000	0.994	0.800	0.889	0.886
5	61	61	45	16	813	16	0.036	0.738	0.981	0.738	0.738	0.718
6	129	135	118	17	670	11	0.034	0.915	0.975	0.874	0.894	0.873
7	42	36	32	4	2384	10	0.006	0.762	0.998	0.889	0.821	0.818
8	52	36	30	6	830	22	0.032	0.577	0.993	0.833	0.682	0.666
9	73	62	47	15	918	26	0.041	0.644	0.984	0.758	0.696	0.675
10	144	113	103	10	716	41	0.059	0.715	0.986	0.912	0.802	0.768
11	118	102	91	11	895	27	0.037	0.771	0.988	0.892	0.827	0.807
12	69	50	43	7	834	26	0.036	0.623	0.992	0.860	0.723	0.704
13	62	56	50	6	790	12	0.021	0.806	0.992	0.893	0.847	0.836
14	50	42	31	11	735	19	0.038	0.620	0.985	0.738	0.674	0.654
15	72	61	56	5	831	16	0.023	0.778	0.994	0.918	0.842	0.830
16	54	39	36	3	693	18	0.028	0.667	0.996	0.923	0.774	0.760
17	56	52	47	5	1251	9	0.011	0.839	0.996	0.904	0.870	0.865
18	89	53	49	4	723	40	0.054	0.551	0.994	0.925	0.690	0.663
19	114	96	85	11	801	29	0.043	0.746	0.986	0.885	0.810	0.785
20	96	92	81	11	863	15	0.027	0.844	0.987	0.880	0.862	0.847
21	66	60	55	5	923	11	0.016	0.833	0.995	0.917	0.873	0.864
22	77	67	56	11	824	21	0.035	0.727	0.987	0.836	0.778	0.759
Total	1698	1496	1278	218	19910	420	0.032	0,748	0,988	0.855	0,793	0,775

Table 2
Performance measures of the trivial methods for EEG arousals detection. Sens = Sensitivity; Spec = Specificity; Prec = Precision.

Solution	Error	Sens	Spec	Prec	F_1 score	Kappa
Always EEG Arousal	0.922	1.000	0.000	0.078	0.144	0.000
Never EEG Arousal	0.078	0.000	1.000	–	0.000	0.000

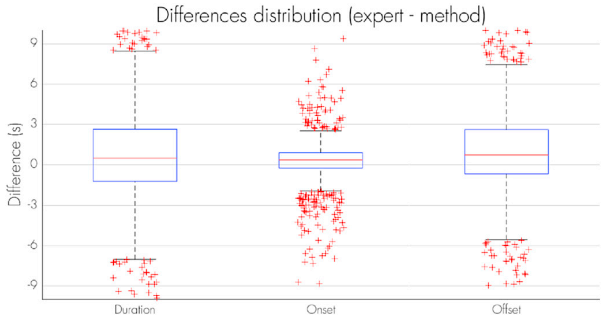


Fig. 5. Distribution of the differences between expert's and method's events.

Table 3
Significant values of the duration, onset and offset pairwise differences (in seconds). Data coverage is calculated using lower and upper adjacent points.

	Lower Adjacent	25th Percentile	Median	75th Percentile	Upper Adjacent	Coverage
Duration	−7.35	−1.68	0.14	2.17	7.89	92.02%
Onset	−1.95	−0.23	0.38	0.93	2.65	88.22%
Offset	−5.44	−0.96	0.39	2.1	6.66	89.82%

be rejected ($p = 0.037$), using a reference significance level of $\alpha = 0.01$. Comparing the onset and offset differences, the null hypothesis that method-expert offset deviations were higher than the onset differences is accepted ($p = 0.959$, Wilcoxon rank sum test). The complementary hypotheses (offset deviations are equal or less equal than onset differences) were both rejected with $p < 0.001$. In other words, the method detects the start of the events more precisely (in accordance with the clinical experts) than their ends. On the other hand, by comparing the time during which both the expert and the method agree in the detection of an event (common overlapping time), against the respective time in disagreement, statistical tests rejected the null hypothesis of equality of medians ($p < 0.001$), while accepted the hypothesis that the median overlapping time is bigger than the non-overlapping time (p = 1). Fig. 6 shows the respective distributions of agreement between expert and method's events. Finally, we also tested the hypothesis that the absolute onset and offset deviations were significantly minor than the common overlapping time. Statistical tests confirmed the result, both for the onset and for the offset times ($p = 1$), rejecting the alternative hypotheses with $p < 0.001$.

4. Discussion

In the presented method preliminary candidate events are first detected by finding abrupt EEG frequency changes in the alpha and beta bands. Even though the official clinical definition does also includes the possibility of frequency shifts to occur in the theta band, in practice the specificity of this band to the occurrence of EEG arousals is rather low. Certainly theta changes are present during the occurrence of arousal events, but transient theta activity does also appear all throughout full-

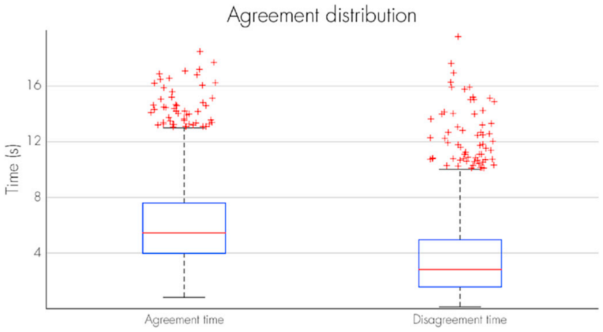


Fig. 6. Distribution of the agreement between expert's and method's events.

night EEG recordings. This activity is not necessarily linked to the occurrence of arousals and thus makes detection very unreliable, specially during periods of drowsy sleep and in stage REM. The identification of arousal activity in the EEG based on changes in the alpha and beta frequency bands can be found in previous works in the literature [11,15,26].

Some empirically established thresholds were used for the parameterization of our method. During the initial detection phase, the use of high threshold values implies the necessity of more abrupt frequency changes to detect an event. But with such configuration, sensitivity toward the less clear cases might be lost. The selected thresholds, therefore, were optimized using the training dataset to maximize the sensitivity-precision trade-off. A possible criticism to the use of thresholds is that by relying on fixed detection values we might not correctly address the signal variability due to subject-dependent characteristics. On this regard, our method does only use relative power and amplitude rates. Thus, we avoid relying on absolute physical measurements, solving the problematic of subject-dependency.

Detection and correction of possible signal artifacts is also of fundamental importance for a robust behavior of the detection algorithm. For this purpose we have incorporated several stages of signal conditioning in our analysis pipeline. In particular, EKG artifact can be seen affecting the EEG and the EMG in some patients. The degree to which this artifact could influence the overall performance is specific of each recording. Normally, in our validation dataset of 22 patients, the differences in performance with or without using the EKG filtering do not change dramatically: overall kappa agreement decreases only to 0.753 from 0.775 when EKG filtering is deactivated. However, for some specific patients in which the EKG artifact is specially relevant, the differences are more clear. For example, recording 2 (see Table 1) shows a drop in the kappa index from 0.747 to 0.636 when filtering is not applied.

Once candidate events are detected, the method reviews each one to find recognizable EEG arousal patterns that match the standard clinical definitions [5]. For this purpose different pattern matching techniques were implemented in which the submental EMG was used as contextual information. Contextual information about sleep and wake periods was also used to discard false positives. In this regard it is worth to mention that the results shown in Table 1 have been obtained when feeding the algorithm with such a previous classification using hypnograms generated by human experts. We have tried a fully automatic approach using a method developed by Alvarez-Estevez et al. [27] for the automatic generation of the hypnogram. The validation results obtained with this approach showed a decay, with a moderate overall kappa of 0.558, a precision and F1-score of 0.742 and 0.586 respectively. It should be noticed that no attempt was made to adapt the aforementioned hypnogram generation algorithm [27] to the characteristics of the current dataset. Thus, these results should be regarded only as illustrative, and future work will be carried out on this direction.

That being said, the overall validation results obtained are satisfactory, specially taking into account the high inter-rater variability between human expert scorers reported in the literature [4,28,29]. Specifically, studies assessing agreement between clinical experts on an epoch-by-epoch EEG arousal scoring task have reported kappa indices in the range 0.47–0.57 [30,31].

The comparison of our results to previous works in the literature is difficult, due to the general lack of a standard benchmark or methodology. Tables 4 and 5 try to summarize the results reported among the relevant related works. Table 4 includes approaches in which only positive event matchings against the clinical reference scoring were counted. On the other hand, Table 5 involves approaches for which a fixed time window (granularity of the time scoring unit differs per approach) was used to compute both agreements and disagreements with respect to the clinical reference scoring.

The metrics reported and the validation methodology differ from each method. Besides, while most of the published approaches in Table 5 keep reporting on the sensitivity and specificity values, we have shown that usually the number of EEG arousals is highly reduced in relation to the total recording time. Thus, the validation procedure in this case should take into account the imbalance of the classes, and report more appropriate measures such as the precision, the F_1 score or the Kappa coefficient.

Table 4
Results reported for methods counting only positive event matchings against the clinical reference; * Values were not explicitly mentioned in the referenced work, but can be derived from the published data.

Method	#Recordings	Sensitivity	Precision	F_1 score	True positive definition
Zamora and Tarassenko [7]	7 (20 min)	0.7–1	0.88–1.0	–	Detections within 10 s of an expert score
De Carli et al. [8]	8	0.88	0.74	–	Overlapping events
Agarwal [11]	2	0.42–0.82	0.57–0.80	0.56–0.77*	Overlapping events

Overall, it can be stated that our method achieves reasonable good results. To be remarked is that, at least for the approximations of Zamora and Tarassenko [7] (Table 4) and of Pacheco and Vaz [6] and Cho et al. [10] (both in Table 5), validation is limited to partial selected periods from the total recording time. In the work of De Carli et al. [8] the standard reference was obtained from a consensus of two human scorers and their own proposed method, which might bias the results. On the other hand, it should be noticed that their method uses as input the expert's annotations of sleep stages to discard false positive detections during stage REM. In Agarwal [11] it is mentioned that the configuration of the method was optimized individually for each of the two testing recordings. Therefore, the validation results might be biased as well. The validation procedure described in Sugi et al. [13] presents a similar problem, as 25% of each recording's data were used for training their model.

The comparison of our results against the works of Shmiel et al. [14], Alvarez-Estevez and Moret-Bonillo [15], Álvarez Estévez [32], Fernández-Varela et al. [17] is specially interesting, given the similarities of the validation approach. On this regard while the sensitivity achieved with our method falls within the range of the reported indices (0.75 vs 0.65–0.88), taking the average of the sensitivity and specificity values, our method outperforms all the rest (see column AUC in Table 5). Moreover, we outperform the four methods with regard to the calculated Precision, F1-score and Kappa metrics. As it has been previously stated, when considering our results using the automatic hypnogram generation method described in Alvarez-Estevez et al. [27], the validation metrics decay, but we obtain values of Precision, F1-score and Kappa within the range of the considered approaches.

Our validation is complemented with statistical analyses on the event's time location as compared to the clinical reference scorings. These analyses have revealed a bias toward better detection of the onset event times. The same analyses have revealed quite good performance in terms of event duration and overlapping time agreement. Actually, we have shown that the onset and offset differences are not statistically relevant in comparison to the common overlapping time.

The analysis of the PSG through the scoring of all the events is costly, due to the complexity of the detection process and the huge amount of data recorded per night. Complex methods would involve long periods of analysis examining the complete PSG to diagnose sleep disorders, which would limit its application in large scale trials. Therefore, to speed up the hole analysis through simple and fast events detection procedures would lead to better times in the diagnosis process. The method proposed to identify arousals accomplishes this goal and achieves good performance in a simple, robust and fast manner.

5. Conclusions

EEG arousals are microstructural events of the sleep that represent

Table 5
Results reported for methods using a fixed time window to compute agreements and disagreements against the clinical reference; * Values were not explicitly mentioned in the referenced work, but can be derived from the published data; AUC = Area Under ROC Curve of one point obtained as (Sensitivity+Specificity)/2; AH = Automatic Hypnogram; NREM = Non-Rapid Eye Movement; TN = True Negative.

Method	#Recordings	Scoring unit	Sensitivity	Specificity	AUC	Precision	F_1 score	Kappa
Pacheco and Vaz [6]	8 (2 h)	30	0.88	–	–	–	–	–
Cho et al. [10]	6 (NREM)	1	0.75	0.93	0.84*	–	–	–
Sugi et al. [13]	8	1.28 (30 for TN)	0.82	0.88	0.85*	–	–	–
Shmiel et al. [14]	20	30	0.75	–	–	0.77	–	–
Alvarez-Estevez and Moret-Bonillo [15]	5	30	0.86	0.77	0.82	0.42*	0.57*	0.44*
Álvarez Estévez [32]	26	30	0.65	0.95	0.80	0.7*	0.68*	0.62*
Fernández-Varela et al. [17]	26	30	0.81	0.88	0.85	0.56*	0.66*	0.58*
Ours	22	30	0.75	0.99	0.87	0.86	0.80	0.78
Ours (AH)	22	30	0.52	0.98	0.75	0.74	0.59	0.56

awakening activity from the EEG. They are associated with sleep fragmentation and therefore their quantification is very important for the diagnosis of different sleep disorders. Visual inspection of the PSG to score these events is complex and very resource-demanding. Given this context, this work presents an automatic detection method with the purpose of helping the clinician in the arousal scoring task. The aim of the proposed solution is to perform a robust detection while relying on the use of well-known, and relatively simple, signal processing techniques. The purpose is to facilitate the described method to be easily implemented on different programming environments.

We concluded that the results achieved in this work are encouraging, yet there is room for improvement. More emphasis will be put in extending the testing dataset for a broader evaluation of the method. To fully automate the method, further work will focus toward automatic differentiation of the sleep and wake periods, with the objective of improving the results reported when our method was used in conjunction with the automatic method for the hypnogram generation [27]. The challenge is to keep a good performance independently of the patient database used as reference.

Conflicts of interest statement

All the authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Acknowledgement

This work has received financial support from the Xunta de Galicia and the European Union (European Regional Development Fund - ERDF) [Grant Code GRC2014/035]; the Spanish Ministerio de Economía y Competitividad (MINECO) and the European Union (ERDF) [Research Project TIN2013-40686P]; the Xunta de Galicia (Centro singular de investigación de Galicia accreditation 2016–2019) and the European Union (ERDF) [Grant Code ED431G/01]; and the Xunta de Galicia and the European Union (European Social Fund - ESF).

References

[1] D.J. Buysse, L. Yu, D.E. Moul, A. Germain, A. Stover, N.E. Dodds, K.L. Johnston, M.A. Shablesky-Cade, P.A. Pilkonis, Development and validation of patient-reported outcome measures for sleep disturbance and sleep-related impairments, Sleep 33 (6) (2010) 781–792.

[2] R. Colter, B. Altevorgt (Eds.), Sleep Disorders and Sleep Deprivation: an Unmet Public Health Problem, Institute of Medicine; Board on Health Sciences Policy, 2006.

[3] C.A. Kushida, M.R. Littner, T. Morgenthaler, C.A. Alessi, D. Bailey, J. Coleman Jr., L. Friedman, M. Hirshkowitz, S. Kapen, M. Kramer, et al., Practice parameters for the indications for polysomnography and related procedures: an update for 2005, Sleep 28 (4) (2005) 499–521.

[4] M. Bonnet, K. Doghramji, T. Roehrs, E. Stepanski, S. Sheldon, A. Walters, M. Wise, A. Chesson, The scoring of arousal in sleep: reliability, validity, and alternatives, J. Clin. Sleep Med. 3 (2) (2007) 133–145.

[5] R.B. Berry, R. Brooks, C.E. Gamaldo, S.M. Harding, C. Marcus, B. Vaughn, The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications, Version 2.3, vol. 1, American Academy of Sleep Medicine, Westchester, IL, 2016.

[6] O. Pacheco, F. Vaz, Integrated system for analysis and automatic classification of sleep EEG, in: Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Biomedical Engineering towards the Year 2000 and beyond (Cat. No.98CH36286), Vol. 4, vol. 20, IEEE, 1998, pp. 2062–2065, http://dx.doi.org/10.1109/IEMBS.1998.747012. ISBN 0-7803-5164-9.

[7] M. Zamora, L. Tarassenko, The study of micro-arousals using neural network analysis of the EEG, in: 9th International Conference on Artificial Neural Networks: ICANN `99, vol. 1999, IEEE, 1999, pp. 625–630, http://dx.doi.org/10.1049/cp:19991180. ISBN 0 85296 721 7.

[8] F. De Carli, L. Nobili, P. Gelcich, F. Ferrillo, A method for the automatic detection of arousals during sleep, Sleep 22 (5) (1999) 561–572. ISSN 0161–8105.

[9] G. Pillar, A. Bar, A. Shlitner, R. Schnall, J. Shefy, P. Lavie, Autonomic arousal index: an automated detection based on peripheral arterial tonometry, Sleep 25 (5) (2002) 543–549. ISSN 0161–8105.

[10] S. Cho, J. Lee, H. Park, K. Lee, Detection of arousals in patients with respiratory sleep disorders using a single channel EEG, in: 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, vol. 3, IEEE, 2005, pp. 2733–2735, http://dx.doi.org/10.1109/IEMBS.2005.1617036. ISBN 0-7803-8741-4, ISSN 1557-170X.

[11] R. Agarwal, Automatic detection of micro-arousals, in: 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, vol. 2, IEEE, 2005, pp. 1158–1161, http://dx.doi.org/10.1109/IEMBS.2005.1616628. ISBN 0-7803-8741-4, ISSN 1557-170X.

[12] U. Malinowska, P. Durka, K. Blinowska, W. Szelenberger, A. Wakarow, Micro- and macrostructure of sleep EEG, IEEE Eng. Med. Biol. Mag. 25 (4) (2006) 26–31, http://dx.doi.org/10.1109/MEMB.2006.1657784. ISSN 0739-5175.

[13] T. Sugi, F. Kawana, M. Nakamura, Automatic EEG arousal detection for sleep apnea syndrome, Biomed. Signal Process. Control 4 (4) (2009) 329–337.

[14] O. Shmiel, T. Shmiel, Y. Dagan, M. Teicher, Data mining techniques for detection of sleep arousals, J. Neurosci. Methods 179 (2) (2009) 331–337.

[15] D. Alvarez-Estevez, V. Moret-Bonillo, Identification of electroencephalographic arousals in multichannel sleep recordings, IEEE Trans. Biomed. Eng. 58 (1) (2011) 54–63, http://dx.doi.org/10.1109/TBME.2010.2075930. ISSN 00189294.

[16] D. Alvarez-Estevez, N. Sánchez-Maroño, A. Alonso-Betanzos, V. Moret-Bonillo, Reducing dimensionality in a database of sleep EEG arousals, Expert Syst. Appl. 38 (2011) 7746–7754.

[17] I. Fernández-Varela, E. Hernández-Pereira, D. Álvarez Estévez, V. Moret-Bonillo, Combining machine learning models for the automatic detection of EEG arousals, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2016.11.086 (In Press) –ISSN 0925–2312.

[18] C.K. Behera, T.K. Reddy, L. Behera, B. Bhattacarya, Artificial neural network based arousal detection from sleep electroencephalogram data, in: 2014 International Conference on Computer, Communications, and Control Technology (I4CT), I4ct, IEEE, 2014, pp. 458–462, http://dx.doi.org/10.1109/I4CT.2014.6914226. ISBN 978-1-4799-4555-9.

[19] D. C. t. Wallant, V. Muto, G. Gaggioni, M. Jaspar, S.L. Chellappa, C. Meyer, G. Vandewalle, P. Maquet, C. Phillips, Automatic artifacts and arousals detection in whole-night sleep EEG recordings, J. Neurosci. Methods 258 (2016) 124–133, http://dx.doi.org/10.1016/j.jneumeth.2015.11.005. ISSN 1872678X.

[20] D. Alvarez-Estevez, A new automatic method for the detection of limb movements and the analysis of their periodicity, Biomed. Signal Process. Control 26 (2016) 117–125, http://dx.doi.org/10.1016/j.bspc.2016.01.008. ISSN 17468094.

[21] V. Alfonso, W. Tompkins, T. Nguyen, S. Luo, ECG beat detection using filter banks, IEEE Trans. Biomed. Eng. 46 (2) (1999) 192–202.

[22] S. Hamilton, Open Source ECG Analysis Software Documentation, Tech. Rep., E.P. Limited, 2002 http://www.eplimited.com/osea13.pdf.

[23] D. Alvarez-Estevez, I. van Velzen, T. Ottolini-Capellen, B. Kemp, Derivation and modeling of two new features for the characterization of rapid and slow eye

movements in electrooculographic sleep recordings, Biomed. Signal Process. Control 35 (2017) 87–99, http://dx.doi.org/10.1016/j.bspc.2017.02.014. ISSN 1746-8094.

[24] B. Kemp, J. Olivan, European data format 'plus' (EDF+), an EDF alike standard format for the exchange of physiological data, Clin. Neurophysiol. 114 (9) (2003) 1755–1761. ISSN 13882457.

[25] J.R. Landis, G.G. Koch, The measurement of observer agreement for categorical data, Biometrics (1977) 159–174.

[26] M. Asyali, R. Berry, M. Khoo, A. Altinok, Determining a continuous marker for sleep depth, Comput. Biol. Med. 37 (11) (2007) 1600–1609.

[27] D. Alvarez-Estevez, J. Fernández-Pastoriza, E. Hernández-Pereira, V. Moret-Bonillo, A method for the automatic analysis of the sleep macrostructure in continuum, Expert Syst. Appl. 40 (2013) 1796–1803.

[28] C. Whitney, D. Gottlieb, S. Redline, R. Norman, R. Dodge, E. Shahar, S. Surovec, F. Nieto, Reliability of scoring respiratory disturbance indices and sleep staging, Sleep 21 (7) (1998) 749–757.

[29] J. Loredo, J. Clausen, S. Ancoli-Israel, J. Dimsdale, Night-to-night arousal variability and interscorer reliability of arousal measurements, Sleep 22 (7) (1999) 916–920.

[30] M. Drinnan, A. Murray, C. Griffiths, G. Gibson, Interobserver variability in recognizing arousal in respiratory sleep disorders, Am. J. Respir. Clin. Care Med. 158 (1998) 358–362.

[31] S. Pittman, M. MacDonald, R. Fogel, A. Malhotra, K. Todros, B. Levy, A. Geva, D. White, Assessment of automated scoring of polysomnographic recordings in a population with suspected sleep-disordered breathing, Sleep 27 (7) (2004) 1394–1403.

[32] D. Álvarez Estévez, Diagnosis of the Sleep Apnea-hypopnea Syndrome. A Comprehensive Approach through an Intelligent System to Support Medical Decision, Ph.D. thesis, Universidade da Coruña, Departamento de Computación, 2012.

2.3 Large-scale validation of an automatic EEG arousal detection algorithm using different heterogeneous databases

- **Title:** Large-scale validation of an automatic EEG arousal detection algorithm using different heterogeneous databases
- **Authors:** Alvarez-Estevez, Diego, Fernández-Varela, Isaac
- **Journal:** Sleep Medicine
- **Editorial:** Elsevier
- **ISSN:**
- **Year:** 2019
- **Volume:**
- **Pages:**
- **DOI:** 10.1016/j.sleep.2019.01.025
- **Available in:** `https://www.sciencedirect.com/science/article/pii/S1389945718303198`

Sleep Medicine 57 (2019) 6–14

Contents lists available at ScienceDirect

Sleep Medicine

journal homepage: www.elsevier.com/locate/sleep

Original Article

Large-scale validation of an automatic EEG arousal detection algorithm using different heterogeneous databases

Diego Alvarez-Estevez [a], [*], Isaac Fernández-Varela [b]

[a] Sleep Center and Clinical Neurophysiology Department, Haaglanden Medisch Centrum, The Hague, The Netherlands
[b] Computer Science Department, University of A Coruña, A Coruña, Spain

ARTICLE INFO

Article history:
Received 25 June 2018
Received in revised form
25 October 2018
Accepted 15 January 2019
Available online 30 January 2019

Keywords:
Automated scoring
EEG arousal
Polysomnography

ABSTRACT

Objective: To assess the validity of an automatic EEG arousal detection algorithm using large patient samples and different heterogeneous databases.
Methods: Automatic scorings were confronted with results from human expert scorers on a total of 2768 full-night PSG recordings obtained from two different databases. Of them, 472 recordings were obtained during a clinical routine at our sleep center and were subdivided into two subgroups of 220 (HMC-S) and 252 (HMC-M) recordings each, according to the procedure followed by the clinical expert during the visual review (semi-automatic or purely manual, respectively). In addition, 2296 recordings from the public SHHS-2 database were evaluated against the respective manual expert scorings.
Results: Event-by-event epoch-based validation resulted in an overall Cohen's kappa agreement of $\kappa = 0.600$ (HMC-S), 0.559 (HMC-M), and 0.573 (SHHS2). Estimated inter-scorer variability on the datasets was, respectively, $\kappa = 0.594$, 0.561 and 0.543. Analyses of the corresponding Arousal Index scores showed associated automatic-human repeatability indices ranges of 0.693–0.771 (HMC-S), 0.646–0.791 (HMC-M), and 0.759–0.791 (SHHS2).
Conclusions: Large-scale validation of our automatic EEG arousal detector on different databases has shown robust performance and good generalization results comparable to the expected levels of human agreement. Special emphasis was put on reproducibility of the results; implementation of our method has been made available online as open source code.

1. Introduction

Electroencephalographic (EEG) arousals are transient events of the sleep EEG indicative of ongoing awakening activity. According to the current clinical reference standards [1] EEG arousals are defined as abrupt shifts in the EEG frequency including alpha, theta and/or frequencies greater than 16 Hz (but not spindles), that last at least 3 s with at least 10 s of stable sleep preceding the change. The scoring of arousals during Rapid Eye Movement (REM) phases requires a concurrent increase in submental electromyogram (EMG) lasting at least 1 s.

Evidence supports EEG arousals as an important component of the sleep process, and their scoring during routine polysomnographic (PSG) examination is essential for evaluating a subject's sleep continuity, and to give treatment and treatment response guidelines to practitioners [2].

Manual visual examination of the entire PSG for the scoring of these events is costly, due to both the complexity and the amount of data involved. Given this context, several studies have explored the possibility of developing automatic analysis software to help clinicians during the scoring process [3–16].

While some of the previous approaches have shown promising performance, validation methods are usually limited to relatively small (ranging from 2 to 31 recordings), controlled, and mostly local and private datasets. It remains a question whether the detection capabilities of these algorithms generalize to larger samples, different databases, and perform well in a clinical (non-controlled) environment. The reality is that the level of acceptability of these algorithms among the clinical community remains low and thus they are rarely used in the clinical practice.

In a recent work [16], we presented a preliminary version of a method for automatic EEG arousal detection, obtaining good validation results on a controlled dataset of 22 PSG in-hospital

[*] Corresponding author. Sleep Center & Clinical Neurophysiology, Haaglanden Medisch Centrum, Lijbaan 32, 2512VA, The Hague, The Netherlands.
E-mail address: diego.alvareze@udc.es (D. Alvarez-Estevez).

https://doi.org/10.1016/j.sleep.2019.01.025
1389-9457/© 2019 Elsevier B.V. All rights reserved.

recordings. Building upon this initial version, an updated algorithm has been developed and set up to work in the clinical environment, where clinicians can choose whether to use it or not as a supportive scoring tool, while reviewing their night recordings.

In this paper, we present a large-scale validation of the performance of this updated approach. Validation has been carried out on a large sample of patients using our sleep center database; we have also extended it by evaluating the algorithm over a large, external, and public database, namely the Sleep Heart Health Study (SHHS) [17]. In each case, the expected level of inter-scorer human variability has been estimated to contextualize the results of the analyses. Moreover, the source code of the algorithm has been published on the internet, making it freely available to the research and clinical communities. To our knowledge, this is the largest and most complete validation ever done of an algorithm of this kind.

2. Methods

2.1. Databases

The validation of the EEG arousal detection algorithm was performed using reference data from two different large databases. The first database is composed of clinical sleep recordings from our sleep center (Haaglanden Medisch Centrum - HMC, The Hague, The Netherlands). Second, an external and publicly accessible source was used, namely the Sleep Heart Health Study (SHHS) database [17]. Each of the databases and the different derived datasets are described in detail below. A summary of resulting demographics and main PSG data are shown in Table 1.

2.1.1. HMC

This data collection is composed of PSG recordings gathered retrospectively from the Haaglanden Medisch Centrum (The Hague, The Netherlands) sleep center database. The PSG recordings were acquired in the course of common clinical practice, and thus did not subject people to any other treatment nor prescribe any additional behavior outside of the usual clinical procedures. Data were anonymized avoiding any possibility of individual patient identification. The study was carried out in full compliance with the corresponding applicable law and under the supervision of the local Medical Ethics Committee. Patient signals were acquired using SOMNOscreen™ plus devices (SOMNOmedics, Germany) and digitized using the EDF + format [18]. PSGs were afterward analyzed offline by clinical experts in the course of common clinical practice. Manual scoring of the events included, among the common standard parameters, the annotation of sleep stages and of EEG arousals. All procedures were performed according to the standard AASM guidelines [1]. As a homogenization criterion, we required the recordings to contain at least 4 h of Total Sleep Time (TST) after clinical scoring. To match a scenario as close as possible to real working conditions, we made no further attempt to filter out, nor to reject any recording, due to specific patient conditions or to poor signal quality. The only required condition was that the recording had been accepted by the clinicians for the manual scoring of EEG arousals. In total, the sample included 472 recordings from patients visiting our center between April and October 2017. Data included both 24-h ambulatory (APSG, n = 352) and in-hospital night (HPSG, n = 143) recordings.

From this database, two separated datasets were arranged as described below.

2.1.1.1. HMC-S. This dataset containing 220 clinical recordings out of the original 472 is composed of 176 APSGs and 45 HPSGs. PSG recordings from this dataset were clinically scored using a semi-automatic approach. First, the clinician used an automatic scoring algorithm for the detection of EEG arousals, and then, on a second pass, the scorer reviewed the results of the automatic scoring replacing, adding or deleting events where necessary. The automatic scoring algorithm used for this purpose was a previous version of the current approach which was described in detail in Ref. [16].

2.1.1.2. HMC-M. This dataset contains 252 clinical recordings out of the original 472, split into 176 APSGs and 98 HPSGs. Recordings from this dataset were scored following the classical clinical routine (ie purely-manual without any support from automatic scoring).

2.1.2. SHHS

The Sleep Heart Health Study (SHHS) is a multi-center cohort study implemented by the National Heart Lung & Blood Institute to determine the cardiovascular and other consequences of sleep-disordered breathing. This database is available online upon permission at the National Sleep Research Resource (NSRR) [19,20]. More information about the rationale, design, and protocol of the SHHS study can be found in the dedicated NSRR section [21] and in the literature [17,22]. A sample of participants who met the SHHS inclusion criteria (age 40 years or older; no history of treatment of sleep apnea; no tracheostomy; no current home oxygen therapy) was invited to participate in the baseline examination of the SHHS, which included an initial polysomnogram (SHHS-1). In all, 6441 individuals were enrolled between November 1, 1995, and January 31, 1998. During exam cycle three (January 2001—June 2003), a second polysomnogram (SHHS-2) was obtained on 3295 of the participants. Raw PSG data are available at NSSR for 5793 subjects within SHHS-1, and 2651 subjects within SHHS-2.

Polysomnograms were obtained in an unattended setting, usually at home, by trained and certified technicians. Specifications for the full montage settings can be found in the corresponding section at the NSRR website [21]. Scoring of sleep stages in SHHS is based on the R&K guidelines [23]. Note that in SHHS, however, no attempt was made to distinguish Stage 3 sleep from Stage 4; both are combined into a single “Deep Sleep” category, similar to current AASM standards [24,1]. Scoring of arousals was done following the ASDA1992 manual [25]. Full specification of all the scoring criteria, as well as quality control procedures for the SHHS study, can be found in the Reading Center Manual of Operations [26].

From SHHS, the SHHS-2 dataset was used as the reference to validate our EEG arousal detection algorithm. Out of the 2561 PSGs available at NSRR, a total of 2492 recordings were selected after excluding those that did not match the general SHHS2 v3 signal montage [27], or for which no attempt to score EEG arousals was performed by the SHHS scorers. From this subset, 60 recordings

Table 1
Summary of demographic data and main PSG characteristics for the different datasets.

Dataset	n	Age	Gender	ArI	AHI
HMC-S	220	52.99 ± 14.33	137 M (62%)/83 F (38%)	12.85 ± 7.86	13.33 ± 15.28
HMC-M	252	51.58 ± 16.22	133 M (53%)/119 F (47%)	12.45 ± 10.48	14.83 ± 21.03
SHHS2	2296	67.41 ± 10.03	1026 M (45%)/1270 F (55%)	12.91 ± 7.02	16.25 ± 15.64

Data are indicated as mean ± standard deviation; n = number of recordings, M = Males, F = Females, ArI = Arousal Index, AHI = Apnea-Hypopnea Index.

were further excluded due to complete technical failure (complete absence of workable EEG and/or EMG signal during the whole recording) resulting in 2433 recordings remaining. No further attempt was made to filter out, or reject any recording, due to poor signal quality conditions. Similarly, as for HMC datasets, the selection excluded recordings for which TST <4 h. In total 2296 recordings were finally included in the validation study of our algorithm.

2.2. *Algorithm overview*

The current version of the EEG arousals detection algorithm is largely based on a previous version which was described and validated elsewhere [16]. The current updated version is an evolution, although it preserves the same original philosophy of simplicity and robustness.

The algorithm works using just one EEG and one EMG chin derivation. The use of an additional ECG channel for the removal of ECG artifacts is optional. The method for ECG artifact removal is based on adaptive filtering, and it has been described in detail elsewhere [28,16].

It is a multistage method that consists first of a signal preprocessing step (digital Notch filtering in both signals, with optional adaptive ECG artifact removal, and high-pass filtering in the EMG), followed by detection of candidate events based on frequency changes in the EEG (power content analysis in the alpha (8–12 Hz) and in the beta (>16 Hz) bands). As a result of these steps candidate arousal regions are identified signaling the presence of EEG arousal activity. The analysis proceeds by pattern matching this activity using individual EEG and EMG relevant features. Eventually, candidate EEG arousals can be merged, or the initial arousal region can be adjusted, usually by extending the corresponding event's offset, if recognizable arousal activity follows. The detection of EEG arousal events involves different subroutines including (i) EEG power-based; (ii) EEG amplitude-based; and (iii) EMG amplitude-based pattern recognition. Finally, false positives are discarded after examining each candidate event within the context of the accepted clinical definitions, including (i) its adequacy in terms of the standard event duration constraints; (ii) the absence of sleep spindle activity; (iii) the absence of EMG activity during REM periods; and (iv) the presence of (at least) 10s of stable sleep preceding the onset of the event.

Adaptations to the algorithm were necessary to support the possibility of different montage configurations, signal sampling rates, and filtering settings (eg, mains interference occurs in America at 60 Hz while at 50 Hz in Europe). Also, detection thresholds were modified to increase the sensitivity for detection of alpha-prevalent arousal events and to achieve better discrimination of sleep spindle activity. Some processing steps were also simplified, namely regarding the EEG power-based pattern recognition (skipping the first false detection check and discarding the whole procedure in the case of alpha arousals) and the detection of concurrent EMG activity. Extended technical information is beyond the scope of this study, and thus the interested reader is referred to the original publication for details [16]. Furthermore, the source code of the algorithm has been made publicly available as open-source, allowing tracking of all the changes and implementation details. The code (implemented using Matlab) can be downloaded from GitHub [29].

2.3. *Experimental procedures*

All the recordings from the datasets described in Section 2.1. Were re-scored by the automatic algorithm. For parameter configuration, two separated and relatively size-reduced datasets, namely HMC-22 and SHHS1-26, were used. Using this approach, it is possible to keep the parameterization independent of the testing data (HMC-S, HMC-M, and SHHS2) therefore allowing the possibility to evaluate the generalization capabilities of the algorithm. HMC-22 was used for the validation of an earlier version of our algorithm; it is composed of 22 in-hospital PSG recordings gathered from the HMC database. A more detailed description of the dataset and the related validation process can be found in Ref. [16]. The SHHS1-26 dataset, on the other hand, is composed of 26 ambulatory PSG recordings gathered from the SHHS-1 study. This dataset was used to validate alternative EEG arousal detection approaches in the past, which are described in detail in Ref. [14] and in Ref. [15]. Subsets of SHHS1-26 were used as well to validate different machine learning-based approaches described in Ref. [12] (n = 20) and in Ref. [13] (n = 10).

Reference derivations for automatic EEG arousal analysis vary per dataset due to differences between the respective clinical montages. Specifically, for HMC datasets a C4/M1 EEG with bipolar submental EMG configuration was used in HPSG recordings, while for APSGs the Cz/O2 EEG derivation was used instead. In both cases, a single-channel modified lead II ECG derivation was used as a reference for the analysis and the removal of ECG artifacts from the EEG and the EMG signals [16,28]. The sampling frequency was 256 Hz for all signals. Out of the two central EEG derivations available for SHHS recordings [21], the C3/A2 channel was used, together with the default bipolar submental EMG trace. Bipolar-lead ECG was sampled at 125 Hz for SHHS-1, and at 250 Hz for SHHS-2 recordings, and it was used analogously for ECG artifact removal purposes.

The analysis included the automatic rescoring of all EEG arousals, while the remaining (non-EEG arousal) expert annotations were left intact. The rescoring includes, for example, the “lights out” and “lights on” markers, and the hypnogram annotations used respectively to determine the valid scoring and the sleeping periods.

To avoid bias in the analysis due to an imbalance in recording times between in-hospital and ambulatory recordings, we considered only Time In Bed (TIB) periods for the validation. Specifically, for HMC datasets this period was extracted straightaway from the “lights out” and “lights on” markers available within each of the EDF + annotation files, set by the scorers while manually reviewing the recordings. For SHHS, such markers were not explicitly available within the file annotations. In this case, TIB was calculated using the variables “*stloutp*” (Lights out time) and “*time_bed*” (TIB in minutes from “lights out” to “lights on”) available within the SHHS2 metadata (for details see Ref. [30]).

Once all the EEG arousals were automatically scored by the algorithm, the validation was performed using two different, complementary approaches. First, event-to-event scoring validation was carried out using a 30 s epoch basis. For this purpose, every EEG arousal event was assigned to a unique epoch, according to the location of its middle point. Using a 2x2 confusion matrix validation metrics for nominal data, namely sensitivity (recall), specificity, precision, F-1 score, and Cohen's kappa index were then calculated.

Second, from a clinical perspective the respective Arousal Index (ArI) scores were calculated and compared per recording. For validation metrics involving numerical data, we used both the Anderson-Darling and the Lilliefors tests, to check the normal distribution hypothesis. In general, statistical testing was conducted using Matlab software, and the reference significance level was set at $\alpha = 0.05$. Correlation coefficients were calculated among the respective automatic and clinical reference ArI scores. Statistical significance for paired differences at the recording level was calculated using the Wilcoxon signed rank test. Moreover, the Intraclass Correlation Coefficient (ICC) was used as the default measure of repeatability to examine scoring differences [31]. Specifically, a two-way absolute single-measures ICC variant of the statistic was considered [32], using the implementation available at [33]. ICC results were calculated both in the original scale and after log-transformation of the respective automatic and clinical reference scores. Furthermore, for non-Gaussian distributions, repeatability was also examined using Generalized Linear Mixed-Effects Models (GLMMs) with log-link and multiplicative overdispersion modeling [34]. Parametric bootstrapping and Bayesian methods were used for interval estimation, and randomization methods were used for significance testing. Results are provided both on the original and the link scales. Specifically, the rptR package [34,35] available for the *R* statistical computing language was used for this purpose.

2.3.1. Assessment of the expected inter-scorer reliability

Analysis of the expected level of inter-scorer variability was performed to adequately contextualize the results of the automatic scoring. For this purpose, a subset of the available PSGs for each dataset was re-scored by an independent expert scorer, and the results were compared against the original clinical scorings. Manual rescoring of all the recordings in each dataset, however, was unattainable in practice due to the high number of recordings, and hence to the high associated costs in human resources. Therefore, a procedure was established to estimate the actual underlying variability using a reduced subset of recordings. The exact procedure is described as follows:

(i) For each dataset, the respective distribution of the kappa indices obtained from the previous automatic vs. clinical validation was used as a reference.
(ii) From this distribution, the five recordings whose associated indices represent the middle of each inter-quartile range, plus the median, were selected as representatives of the whole population. That is, for each dataset, the recordings with kappa scores on the 12.5, 37.5, 50, 62.5 and 87.5 percentiles were used.
(iii) Each of these recordings was then re-scored by a dedicated expert scorer (not present during the first scoring round), blinded to the results of the original analysis. Display montages at the rescoring step were configured to match the same conditions as during the original clinical scoring. Note that during the rescoring, the hypnogram that resulted from the first scoring was available for contextual interpretation. Its modification, on the other hand, was not allowed. Note as well that respiratory activity traces were omitted from the display for EEG arousal scoring.

Analysis of the rescoring results was carried out by calculating the corresponding derived kappa indices, and by confronting them to both the corresponding clinical and automatic results. Further statistical analyses of the corresponding ArI indices were omitted; as with five measurements per dataset, a low statistical power of the derived metrics was expected [36].

3. Results

Results of epoch-based event-by-event validation are summarized in Tables 2 and 3. In Table 2, and for each dataset, the total number of epochs and the respective validation metrics are accumulated across all the recordings. In Table 3 statistical descriptors are shown by considering the respective per-recording distributions. In general, indices do not follow a normal distribution, and thus data is presented using the median and the associated first and third quartiles.

For the SHHS2 dataset Fig. 1 shows the corresponding kappa index distributions in relation to the respective signal quality scores (check [26] for details on SHHS quality assessment procedures). Kruskal–Wallis analyses resulted in $p = 0.004$ and $p < 0.001$ respectively for the EEG and chin EMG distributions. Subsequent multiple comparison tests under the Tukey's significant difference criterion showed, however, no group differences for the EEG, while for the EMG, only group 1 was significantly different from groups 4 and 5. Unfortunately, quality assessment data were not available for the HMC database to perform a similar analysis.

3.1. Statistical analysis of diagnostic indices

The results from the corresponding ArI distribution analyses are shown in Table 4. Distribution descriptors using the mean and standard derivation, as well as the median and the respective interquartile ranges, are shown per dataset. Individual and difference distributions were analyzed showing non-Gaussian distributions in general ($p < 0.01$ in all the cases). For the difference distributions, the corresponding Wilcoxon paired test p-value is explicitly shown in the last column. In this respect, results reject the null hypothesis H_0 “median of differences is zero” at $\alpha = 0.05$, but for the HMC-M dataset ($p = 0.224$). A further tailed analysis shows that differences for HMC-S and SHHS2 are not significant anymore when assuming a median difference bias of $+0.3$ ($p = 0.088$ and $p = 0.104$ respectively).

Results of the repeatability analyses are shown in Table 5. The linear correlation coefficient (r) and the ICC indices are shown, both for the original and for the log-transformed variables. GLMMs are adequate in the case of non-Gaussian distributions [34], and the corresponding derived indices were similarly calculated in the original and the latent scales. In both cases, a log-link function was used. Notice that reporting repeatability of the transformed variables is the most interesting choice in most of the cases [34,37]. For completeness, however, here both estimates are reported. In all the cases statistical significance of the respective tests was confirmed ($p < 0.001$ for all the indices).

3.2. Expected inter-rater variability analysis

Results from the expected inter-scorer variability analysis are shown in Table 6. For each recording, the representative percentile within its dataset and the time spent during the manual rescoring are indicated, respectively, in columns 2 and 3. The resulting kappa indices are also respectively reported for the manual rescoring vs. the original clinical annotations (R–C, column 4), the automatic vs. the original clinical annotations (A-C, column 5), and the automatic vs. the manual rescoring (A-R, column 6) analyses. From these three, R–C is considered to set the reference for the expected levels

Table 2
Overall results of the event-by-event epoch-based validation on the testing datasets.

Dataset	#Epochs	TP	FP	TN	FN	Sens	Spec	Prec	F1-score	Kappa
HMC-S	207312	12492	5170	180600	9050	0.580	0.972	0.707	0.637	0.600
HMC-M	236336	13130	7340	205668	10198	0.563	0.953	0.641	0.600	0.559
SHHS2	2201487	119702	41398	1928384	112003	0.517	0.979	0.743	0.610	0.573

Sensitivity (Sens), Specificity (Spec), Precision (Prec), F1-score and Cohen's kappa index are calculated based on the total number of cases in the respective contingency table. For each dataset the total number of epochs are accumulated across all the recordings; TP = True Positives, FP = False Negatives, TN = True Negatives, FN = False Negatives.

Table 3
Distribution descriptors of the per-recording event-by-event validation metrics.

Dataset	Sens	Spec	Prec	F1-score	Kappa
HMC-S	0.608 (0.476, 0.724)	0.978 (0.966, 0.986)	0.715 (0.618, 0.804)	0.643 (0.539, 0.712)	0.609 (0.494, 0.683)
HMC-M	0.587 (0.438, 0.731)	0.973 (0.954, 0.987)	0.667 (0.484, 0.784)	0.571 (0.489, 0.658)	0.529 (0.435, 0.621)
SHHS2	0.509 (0.394, 0.634)	0.983 (0.973, 0.989)	0.757 (0.661, 0.824)	0.590 (0.503, 0.683)	0.552 (0.461, 0.651)

Data is shown as Q2 (Q1, Q3) quartiles; Sens = Sensitivity, Spec = Specificity, Prec = Precision.

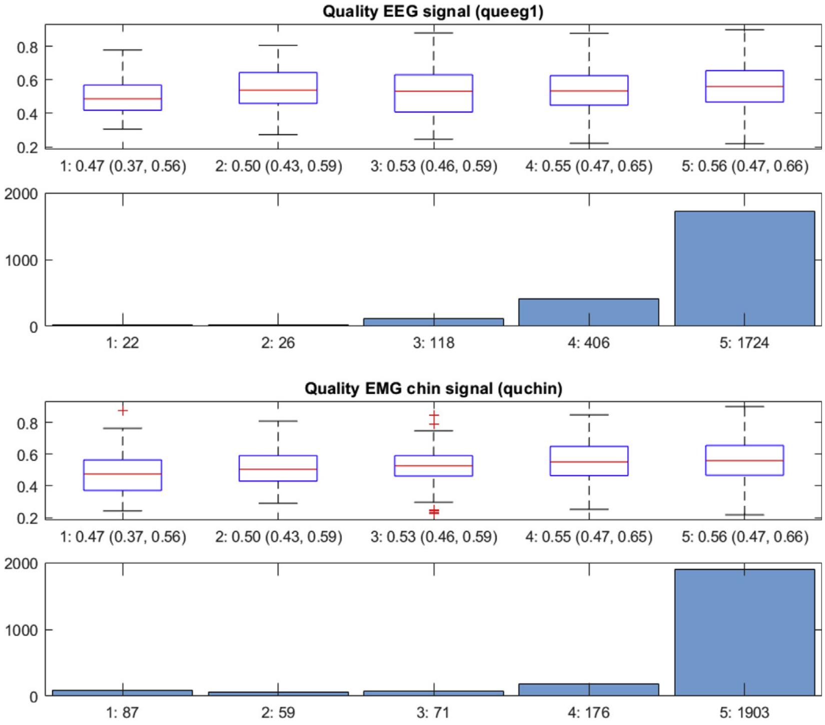


Fig. 1. Signal quality assessments for the 2296 recordings used for validation in the SHHS2 dataset (automatic vs. manual event-by-event validation). Grades were assigned by SHHS scorers according to the SHHS quality assessment procedures. In SHHS2 values vary from 1 (poorest) to 5 (best) and reflect the proportion of sleep time in which the signals were free of artifact; "1": <25%, "2": 25–49%, "3": 50–74%, "4": 75–94%, "5": >95%. Upper plot: "Quality of the EEG signal (queeg1)". Lower plot: "Quality of the EMG chin signal (quchin)." In each case, the first subplot shows the corresponding kappa distributions per group (numerical values for the median and the inter-quartile ranges are indicated below). The subsequent subplot shows a histogram with the number of recordings involved in the corresponding category.

of (human) inter-scorer variability, as this is the one involving the two independent manual scorings.

By comparing the average kappa values for R–C and A-C similar ranges are noticed (columns 4 and 5: HMC-S 0.594/0.595, HMC-M 0.561/0.523, SHHS2 0.552/0.543); that supports the hypothesis of the automatic algorithm behaving as "one expert more" (ie, no important differences are evidenced between the human–human and the automatic-human scorings regarding kappa agreement). There is a slight global increase in the A-R agreements (column 6: 0.602, 0.686, 0.564) compared to the respective A-C (column 5: HMC-S 0.595, HMC-M 0.523, SHHS2 0.552) and R–C (column 4: HMC-S 0.594, HMC-M 0.561, SHHS2 0.543) agreements. This might be indicative of the automatic algorithm behaving more as "the rescoring expert" than as "the original clinical scorers." The effect is more noticeable among the HMC-M recordings. However, it is seldom appreciable for HMC-S and SHHS2 for that to be considered an effective global bias.

Accounting to the differences between the HMC-S and the HMC-M datasets, we might speculate about an expected increment in the variability values in the second case. This is because scoring on the

Table 4
Summary of statistical tests for the diagnostic ArI indices (automatic vs. clinical reference).

Dataset	Individual distributions				Difference distribution		
	Reference		Auto		Ref – Auto		Wilcoxon paired test
	Mean ± SD	Q2 (Q1, Q3)	Mean ± SD	Q2 (Q1, Q3)	Mean ± SD	Q2 (Q1, Q3)	*p*-value
HMC-S	13.32 ± 8.01**	11.86 (7.68, 16.61)	12.47 ± 8.06**	10.74 (7.42, 15.03)	0.84 ± 5.41**	0.40 (−1.99, 3.97)	0.023
HMC-M	12.45 ± 10.48**	9.97 (5.46, 15.52)	12.97 ± 10.14**	10.56 (6.92, 15.14)	−0.52 ± 6.68*	−0.60 (−4.21, 3.50)	0.224
SHHS2	12.91 ± 7.02**	11.54 (8.13, 15.97)	12.56 ± 7.73**	10.74 (7.37, 15.49)	0.35 ± 4.89**	0.44 (−2.07, 3.02)	p < 0.001

*Normality test rejected with p < 0.01; **normality test rejected with p < 0.001.

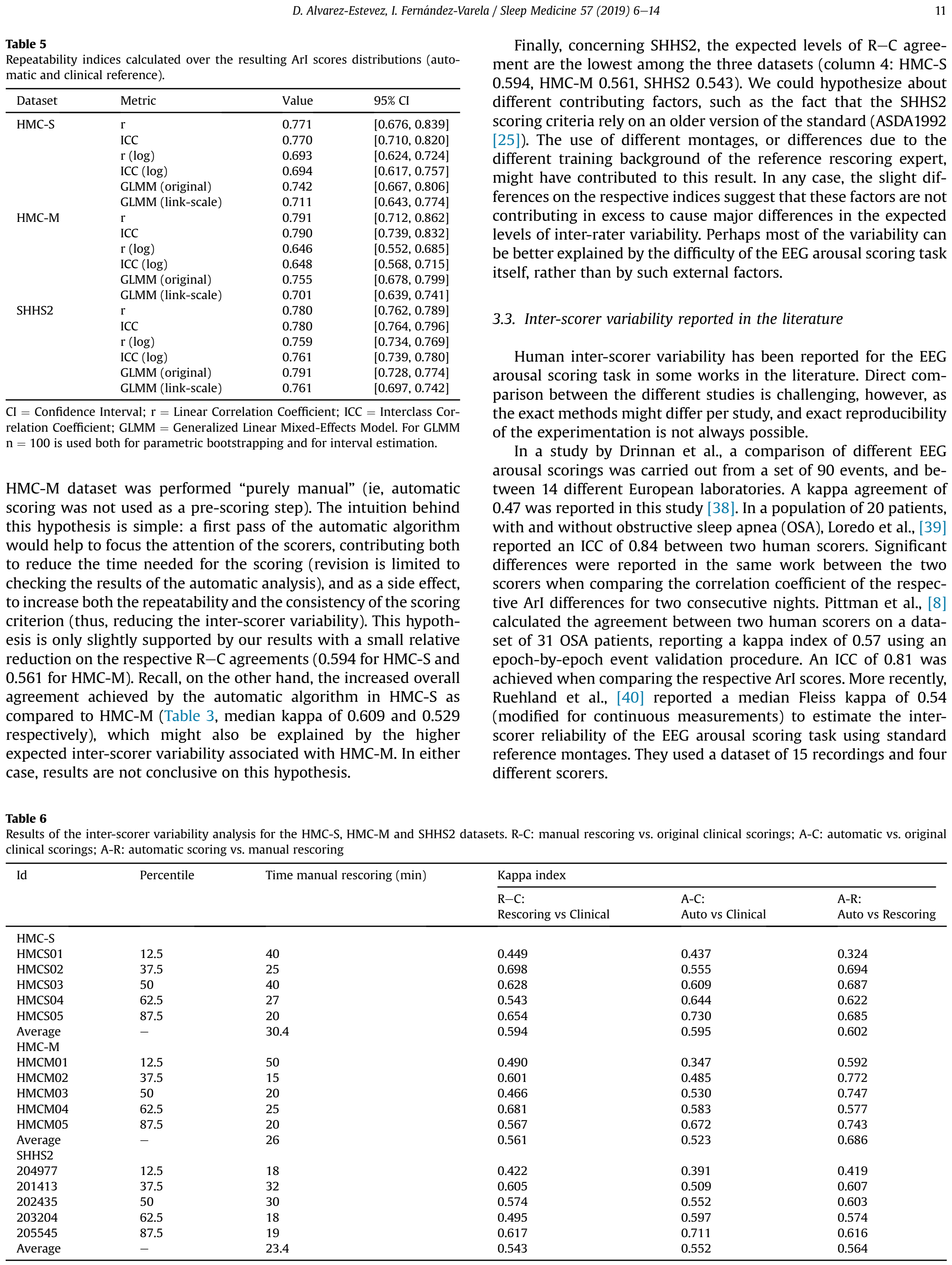


Table 5
Repeatability indices calculated over the resulting ArI scores distributions (automatic and clinical reference).

Dataset	Metric	Value	95% CI
HMC-S	r	0.771	[0.676, 0.839]
	ICC	0.770	[0.710, 0.820]
	r (log)	0.693	[0.624, 0.724]
	ICC (log)	0.694	[0.617, 0.757]
	GLMM (original)	0.742	[0.667, 0.806]
	GLMM (link-scale)	0.711	[0.643, 0.774]
HMC-M	r	0.791	[0.712, 0.862]
	ICC	0.790	[0.739, 0.832]
	r (log)	0.646	[0.552, 0.685]
	ICC (log)	0.648	[0.568, 0.715]
	GLMM (original)	0.755	[0.678, 0.799]
	GLMM (link-scale)	0.701	[0.639, 0.741]
SHHS2	r	0.780	[0.762, 0.789]
	ICC	0.780	[0.764, 0.796]
	r (log)	0.759	[0.734, 0.769]
	ICC (log)	0.761	[0.739, 0.780]
	GLMM (original)	0.791	[0.728, 0.774]
	GLMM (link-scale)	0.761	[0.697, 0.742]

CI = Confidence Interval; r = Linear Correlation Coefficient; ICC = Interclass Correlation Coefficient; GLMM = Generalized Linear Mixed-Effects Model. For GLMM n = 100 is used both for parametric bootstrapping and for interval estimation.

HMC-M dataset was performed "purely manual" (ie, automatic scoring was not used as a pre-scoring step). The intuition behind this hypothesis is simple: a first pass of the automatic algorithm would help to focus the attention of the scorers, contributing both to reduce the time needed for the scoring (revision is limited to checking the results of the automatic analysis), and as a side effect, to increase both the repeatability and the consistency of the scoring criterion (thus, reducing the inter-scorer variability). This hypothesis is only slightly supported by our results with a small relative reduction on the respective R–C agreements (0.594 for HMC-S and 0.561 for HMC-M). Recall, on the other hand, the increased overall agreement achieved by the automatic algorithm in HMC-S as compared to HMC-M (Table 3, median kappa of 0.609 and 0.529 respectively), which might also be explained by the higher expected inter-scorer variability associated with HMC-M. In either case, results are not conclusive on this hypothesis.

Finally, concerning SHHS2, the expected levels of R–C agreement are the lowest among the three datasets (column 4: HMC-S 0.594, HMC-M 0.561, SHHS2 0.543). We could hypothesize about different contributing factors, such as the fact that the SHHS2 scoring criteria rely on an older version of the standard (ASDA1992 [25]). The use of different montages, or differences due to the different training background of the reference rescoring expert, might have contributed to this result. In any case, the slight differences on the respective indices suggest that these factors are not contributing in excess to cause major differences in the expected levels of inter-rater variability. Perhaps most of the variability can be better explained by the difficulty of the EEG arousal scoring task itself, rather than by such external factors.

3.3. *Inter-scorer variability reported in the literature*

Human inter-scorer variability has been reported for the EEG arousal scoring task in some works in the literature. Direct comparison between the different studies is challenging, however, as the exact methods might differ per study, and exact reproducibility of the experimentation is not always possible.

In a study by Drinnan et al., a comparison of different EEG arousal scorings was carried out from a set of 90 events, and between 14 different European laboratories. A kappa agreement of 0.47 was reported in this study [38]. In a population of 20 patients, with and without obstructive sleep apnea (OSA), Loredo et al., [39] reported an ICC of 0.84 between two human scorers. Significant differences were reported in the same work between the two scorers when comparing the correlation coefficient of the respective ArI differences for two consecutive nights. Pittman et al., [8] calculated the agreement between two human scorers on a dataset of 31 OSA patients, reporting a kappa index of 0.57 using an epoch-by-epoch event validation procedure. An ICC of 0.81 was achieved when comparing the respective ArI scores. More recently, Ruehland et al., [40] reported a median Fleiss kappa of 0.54 (modified for continuous measurements) to estimate the inter-scorer reliability of the EEG arousal scoring task using standard reference montages. They used a dataset of 15 recordings and four different scorers.

Table 6
Results of the inter-scorer variability analysis for the HMC-S, HMC-M and SHHS2 datasets. R-C: manual rescoring vs. original clinical scorings; A-C: automatic vs. original clinical scorings; A-R: automatic scoring vs. manual rescoring

Id	Percentile	Time manual rescoring (min)	Kappa index		
			R–C: Rescoring vs Clinical	A-C: Auto vs Clinical	A-R: Auto vs Rescoring
HMC-S					
HMCS01	12.5	40	0.449	0.437	0.324
HMCS02	37.5	25	0.698	0.555	0.694
HMCS03	50	40	0.628	0.609	0.687
HMCS04	62.5	27	0.543	0.644	0.622
HMCS05	87.5	20	0.654	0.730	0.685
Average	–	30.4	0.594	0.595	0.602
HMC-M					
HMCM01	12.5	50	0.490	0.347	0.592
HMCM02	37.5	15	0.601	0.485	0.772
HMCM03	50	20	0.466	0.530	0.747
HMCM04	62.5	25	0.681	0.583	0.577
HMCM05	87.5	20	0.567	0.672	0.743
Average	–	26	0.561	0.523	0.686
SHHS2					
204977	12.5	18	0.422	0.391	0.419
201413	37.5	32	0.605	0.509	0.607
202435	50	30	0.574	0.552	0.603
203204	62.5	18	0.495	0.597	0.574
205545	87.5	19	0.617	0.711	0.616
Average	–	23.4	0.543	0.552	0.564

In children (n = 36) with and without OSA, Wong et al., [41] calculated the differences between two human scorers resulting in an overall ICC of 0.90 (0.88 for the normal group).

It is worth mentioning the study of Whitney et al., [42] as it concerns human variability analysis in the SHHS database. In their study, a subset of 30 recordings was used reporting an ICC of 0.54 between three different scorers. The ICC agreement increased to 0.72 when the two most experienced scorers were compared. The intra-scorer variability was also analyzed on 20 out of the 30 recordings, and significant differences were found for two out of the three scorers using a paired t-test among the respective ArI derived indices [42].

The highest inter-rater agreement published in the literature can be found in the study of Smurra et al. [43], who analyzed both the inter- and the intra-scorer reliability for two different scorers on a set of 20 OSA patients. Their analysis was carried out according to two different scoring standards, namely the ASDA1992 and the ULC. In their work Smurra et al. reported an inter-scorer ICC of 0.96 for the ASDA1992 and 0.98 for the UCL standards. Moreover, no statistical differences were found using one-way ANOVA analysis, when evaluating intra-scorer variabilities between the two scoring references [43].

3.4. Results from other automatic approaches in the literature

Some previous works have performed validation procedures based on positive (overlapping) event matchings against the clinician's scorings (Zamora and Tarassenko [5], De Carli et al. [4], and Agarwal [7]). Sensitivity and precision values vary on these studies (0.42–1.00 and 0.57–1.00, respectively) as well as the number of PSGs involved (from two to eight).

Some other works have carried out epoch-by-epoch validation procedures using time-fixed scoring units. Cho et al., [6] used 1s epochs reporting a sensitivity of 0.75 and a specificity of 0.93 on a set of six recordings. Sugi et al. [10], on the other hand, used a 1.28s epoch length for counting positive matches, while a 30s time reference was used for the computation of true negative scores. Reported indices of sensitivity and specificity were 0.82 and 0.88, respectively, on a dataset of eight recordings. Using a 30s time reference, as in our study, the method of Pacheco and Vaz [3] achieved a sensitivity of 0.88 on selected 2-h periods from eight PSG recordings. In another study [11], using a dataset of 20 full recordings and a 30s epoch reference, Shmiel et al. obtained a sensitivity of 0.75 and a precision of 0.77.

We note that, at least for the approximations of Zamora and Tarassenko [5], Pacheco and Vaz [3], and Cho et al. [6], the validation was limited to partial pre-selected periods out of the total recording time. In the work of De Carli et al. [4], the standard reference was obtained from the consensus of two human scorers and their proposed method, which might bias the result. Agarwal [7] mentioned that the configuration of the method was optimized individually for each of the two testing recordings. Therefore, the validation results might be biased as well. The validation procedure described in Sugi et al., [10] presents a similar problem, as 25% of each recording's data were used for training their model. Notice as well that none of the previous works have reported on the respective expected values of inter-scorer reliability for their datasets.

The work of Pittman et al., [8] is notable in this regard. Using a dataset of 31 recordings and following a similar 30s epoch-by-epoch validation procedure, they reported a kappa index of 0.57 between two reference human scorers. Their automatic algorithm achieved kappa values of 0.28 and 0.30, respectively, against each of the two human scorers. In the same work, also, a comparison of the ArI derived indices was performed, resulting in a human–human ICC of 0.81, and a human-computer ICC of 0.58 and 0.72.

3.5. Comparison with previous automatic approaches from the authors

As introduced in Section 2.3 the authors have attempted automatic EEG arousal scoring in the past following different approaches [16,15,14,13,12].

Specifically, in Ref. [14] and in Ref. [15], the full SHHS1-26 dataset was used, and in Ref. [16], the HMC-22 dataset was taken as reference for the validation of a preliminary version of the current algorithm. Table 7 shows the results of the current version of the algorithm using the HMC-22 and the SHHS1-26 datasets, together with the results obtained in the original publications. On each case validation procedures were replicated following the same conditions as in the original studies, therefore allowing one-by-one direct comparison of the results. The total number of scorable 30s epochs in HMC-22 is of 21,826 and in SHHS1-26 of 31,080.

To set a baseline to evaluate the generalization capabilities of the new version of the algorithm with respect the earlier version presented in Ref. [16], the original version was re-evaluated using the SHHS1-26 dataset. Also, we were able to run the algorithm described in Ref. [14] (originally validated for the SHHS1-26 dataset only) using the HMC-22 dataset. Similar attempts to re-run the algorithm described in Ref. [15] in the HMC-22 dataset were unfortunately unsuccessful, as considerable recoding effort would have been necessary to enable the analysis with the alternative database.

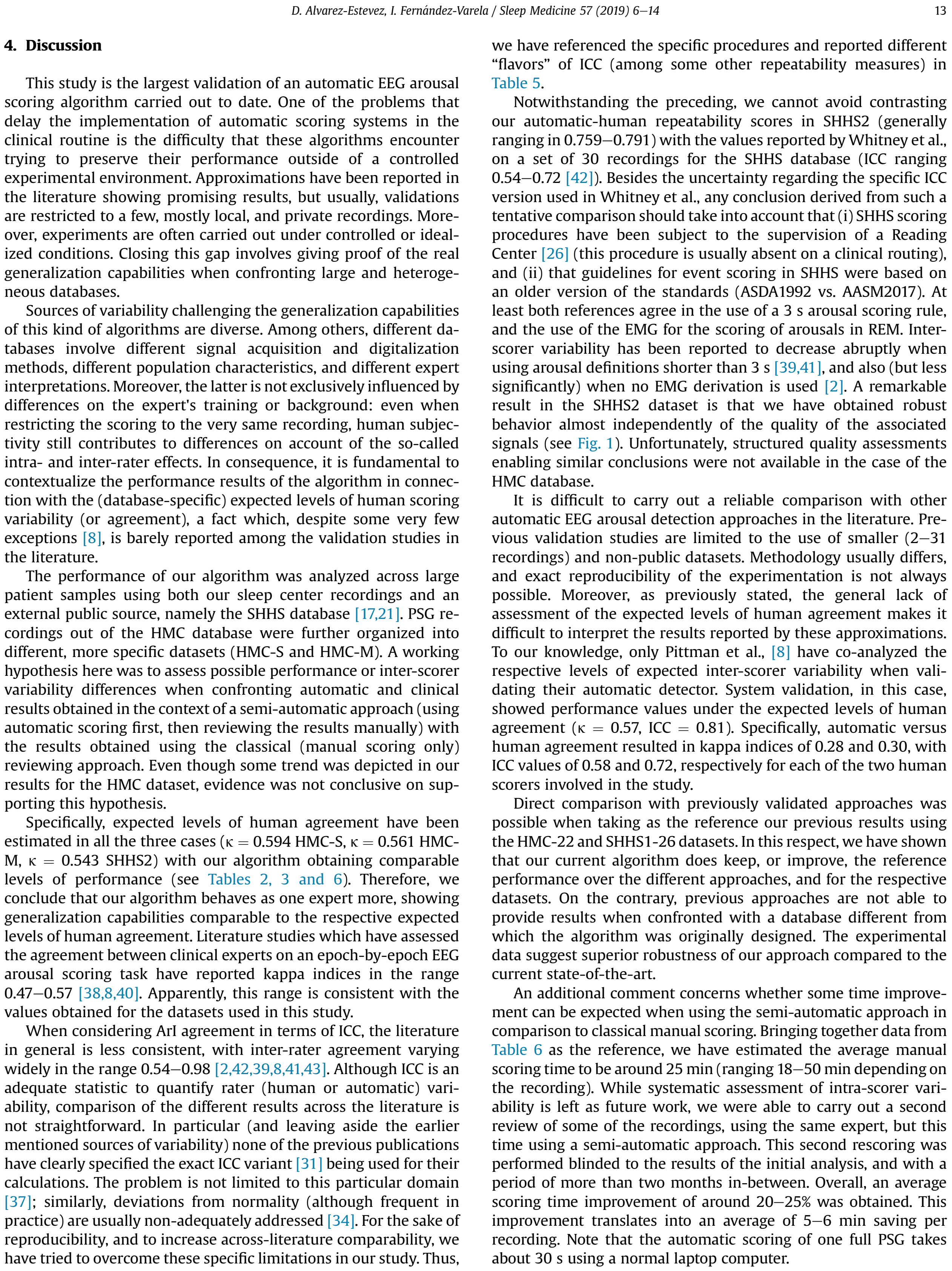
From Table 7 we observe that the current version of the algorithm considerably outperforms its predecessor [16] in SHHS1-26 while keeping a similar performance in HMC-22. The current version also outperforms, in general, all of the remaining examined approaches, namely [14,15]. Given that the SHHS1-26 and the HMC-22 datasets were used during the development phase of the present algorithm, arguably there might be the presence of bias in these results. Nevertheless, the generalization capabilities of the current version of the algorithm have been already proven using the large and independent HMC-S, HMC-M and SHHS2 datasets. Thus, the improvement in the performance is rather interpreted as confirming evidence of the good dataset generalization capabilities of the updated version.

A lack-of-generalization effect can be observed regarding the results of the algorithm described in Ref. [14], showing a decay in the performance when reexamined in HMC-22. Hence, this result supports the hypothesis that the original performance for the method described in Ref. [14], evaluated in the SHHS1-26 dataset, included, at least, a component of database-specific overfitting. Overall, the results show that the previously reported algorithms ultimately do not generalize well when confronted with a change in the database source.

Table 7
Performance comparison of previous automatic approaches on the alternative SHHS1-26 and HMC-22 datasets.

SHHS1-26	Sens	Spec	Prec	F1-score	Kappa
Current	0.581	0.979	0.739	0.634	0.597
Approach in Ref. [16]	0.329	0.992	0.830	0.450	0.405
Approach in Ref. [14]	0.656	0.949	0.649	0.629	0.573
Approach in Ref. [15]	0.810	0.878	0.560	0.660	0.580
HMC-22					
Current	0.791	0.981	0.807	0.792	0.773
Approach in Ref. [16]	0.748	0.988	0.855	0.793	0.775
Approach in Ref. [14]	0.531	0.962	0.555	0.506	0.470

Following the format used in the original publications, results are shown averaging the respective per-patient indices, calculated over the whole recording time; Sens = Sensitivity, Spec = Specificity, Prec = Precision.



4. Discussion

This study is the largest validation of an automatic EEG arousal scoring algorithm carried out to date. One of the problems that delay the implementation of automatic scoring systems in the clinical routine is the difficulty that these algorithms encounter trying to preserve their performance outside of a controlled experimental environment. Approximations have been reported in the literature showing promising results, but usually, validations are restricted to a few, mostly local, and private recordings. Moreover, experiments are often carried out under controlled or idealized conditions. Closing this gap involves giving proof of the real generalization capabilities when confronting large and heterogeneous databases.

Sources of variability challenging the generalization capabilities of this kind of algorithms are diverse. Among others, different databases involve different signal acquisition and digitalization methods, different population characteristics, and different expert interpretations. Moreover, the latter is not exclusively influenced by differences on the expert's training or background: even when restricting the scoring to the very same recording, human subjectivity still contributes to differences on account of the so-called intra- and inter-rater effects. In consequence, it is fundamental to contextualize the performance results of the algorithm in connection with the (database-specific) expected levels of human scoring variability (or agreement), a fact which, despite some very few exceptions [8], is barely reported among the validation studies in the literature.

The performance of our algorithm was analyzed across large patient samples using both our sleep center recordings and an external public source, namely the SHHS database [17,21]. PSG recordings out of the HMC database were further organized into different, more specific datasets (HMC-S and HMC-M). A working hypothesis here was to assess possible performance or inter-scorer variability differences when confronting automatic and clinical results obtained in the context of a semi-automatic approach (using automatic scoring first, then reviewing the results manually) with the results obtained using the classical (manual scoring only) reviewing approach. Even though some trend was depicted in our results for the HMC dataset, evidence was not conclusive on supporting this hypothesis.

Specifically, expected levels of human agreement have been estimated in all the three cases ($\kappa = 0.594$ HMC-S, $\kappa = 0.561$ HMC-M, $\kappa = 0.543$ SHHS2) with our algorithm obtaining comparable levels of performance (see Tables 2, 3 and 6). Therefore, we conclude that our algorithm behaves as one expert more, showing generalization capabilities comparable to the respective expected levels of human agreement. Literature studies which have assessed the agreement between clinical experts on an epoch-by-epoch EEG arousal scoring task have reported kappa indices in the range 0.47–0.57 [38,8,40]. Apparently, this range is consistent with the values obtained for the datasets used in this study.

When considering ArI agreement in terms of ICC, the literature in general is less consistent, with inter-rater agreement varying widely in the range 0.54–0.98 [2,42,39,8,41,43]. Although ICC is an adequate statistic to quantify rater (human or automatic) variability, comparison of the different results across the literature is not straightforward. In particular (and leaving aside the earlier mentioned sources of variability) none of the previous publications have clearly specified the exact ICC variant [31] being used for their calculations. The problem is not limited to this particular domain [37]; similarly, deviations from normality (although frequent in practice) are usually non-adequately addressed [34]. For the sake of reproducibility, and to increase across-literature comparability, we have tried to overcome these specific limitations in our study. Thus, we have referenced the specific procedures and reported different "flavors" of ICC (among some other repeatability measures) in Table 5.

Notwithstanding the preceding, we cannot avoid contrasting our automatic-human repeatability scores in SHHS2 (generally ranging in 0.759–0.791) with the values reported by Whitney et al., on a set of 30 recordings for the SHHS database (ICC ranging 0.54–0.72 [42]). Besides the uncertainty regarding the specific ICC version used in Whitney et al., any conclusion derived from such a tentative comparison should take into account that (i) SHHS scoring procedures have been subject to the supervision of a Reading Center [26] (this procedure is usually absent on a clinical routing), and (ii) that guidelines for event scoring in SHHS were based on an older version of the standards (ASDA1992 vs. AASM2017). At least both references agree in the use of a 3 s arousal scoring rule, and the use of the EMG for the scoring of arousals in REM. Interscorer variability has been reported to decrease abruptly when using arousal definitions shorter than 3 s [39,41], and also (but less significantly) when no EMG derivation is used [2]. A remarkable result in the SHHS2 dataset is that we have obtained robust behavior almost independently of the quality of the associated signals (see Fig. 1). Unfortunately, structured quality assessments enabling similar conclusions were not available in the case of the HMC database.

It is difficult to carry out a reliable comparison with other automatic EEG arousal detection approaches in the literature. Previous validation studies are limited to the use of smaller (2–31 recordings) and non-public datasets. Methodology usually differs, and exact reproducibility of the experimentation is not always possible. Moreover, as previously stated, the general lack of assessment of the expected levels of human agreement makes it difficult to interpret the results reported by these approximations. To our knowledge, only Pittman et al., [8] have co-analyzed the respective levels of expected inter-scorer variability when validating their automatic detector. System validation, in this case, showed performance values under the expected levels of human agreement ($\kappa = 0.57$, ICC $= 0.81$). Specifically, automatic versus human agreement resulted in kappa indices of 0.28 and 0.30, with ICC values of 0.58 and 0.72, respectively for each of the two human scorers involved in the study.

Direct comparison with previously validated approaches was possible when taking as the reference our previous results using the HMC-22 and SHHS1-26 datasets. In this respect, we have shown that our current algorithm does keep, or improve, the reference performance over the different approaches, and for the respective datasets. On the contrary, previous approaches are not able to provide results when confronted with a database different from which the algorithm was originally designed. The experimental data suggest superior robustness of our approach compared to the current state-of-the-art.

An additional comment concerns whether some time improvement can be expected when using the semi-automatic approach in comparison to classical manual scoring. Bringing together data from Table 6 as the reference, we have estimated the average manual scoring time to be around 25 min (ranging 18–50 min depending on the recording). While systematic assessment of intra-scorer variability is left as future work, we were able to carry out a second review of some of the recordings, using the same expert, but this time using a semi-automatic approach. This second rescoring was performed blinded to the results of the initial analysis, and with a period of more than two months in-between. Overall, an average scoring time improvement of around 20–25% was obtained. This improvement translates into an average of 5–6 min saving per recording. Note that the automatic scoring of one full PSG takes about 30 s using a normal laptop computer.

To conclude, we would like to refer to the minimalist nature of the algorithm with regard to the number of signals involved in the analysis. Specifically, our automatic algorithm operates using one EEG and one chin EMG channels. The choice of the specific EEG and EMG derivations used in this study was driven by the availability of the respective database montages. Central EEG derivations (when possible, referenced to the mastoid, and otherwise to the occipital regions) were chosen in HMC to match as close as possible the respective SHHS montages. Performance effects on HMC by the choice of different EEG channels have not been assessed. While we have opted to use an additional ECG trace for ECG artifact removal, unpublished data show that, when evaluated on a large patient sample, ECG filtering does not contribute significantly to the overall performance of our algorithm. However, it does successfully address some specific subsets of recordings highly affected by ECG intrusion.

Future work might also explore the extension of the current method to support multichannel EEG. On an earlier study using a different approach, we have shown that further improvement could be expected by combining independent information from different channels [15].

Acknowledgments

This work has received financial support from the Xunta de Galicia and the European Union (European Social Fund — ESF)

Conflict of interest

None.

The ICMJE Uniform Disclosure Form for Potential Conflicts of Interest associated with this article can be viewed by clicking on the following link: https://doi.org/10.1016/j.sleep.2019.01.025.

References

[1] Berry R, Brooks R, Gamaldo C, et al. The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specifications, version 2.3, vol. 1. Westchester, IL: American Academy of Sleep Medicine; 2016.

[2] Bonnet M, Doghramji K, Roehrs T, et al. The scoring of arousal in sleep: reliability, validity, and alternatives. J Clin Sleep Med 2007;3(2):133–45.

[3] Pacheco O, Vaz F. Integrated system for analysis and automatic classification of sleep EEG. In: 20th annual international conference of the IEEE engineering in medicine and biology society, Hong Kong, China; 1998.

[4] De Carli F, Nobili L, Gelcich P, et al. A method for the automatic detection of arousals during sleep. Sleep 1999;22(5):561–72.

[5] Zamora M, Tarassenko L. The study of micro-arousals using neural network analysis of the EEG. In: 9th international conference on artificial neural networks. Edinburgh, UK: ICANN'99); 1999.

[6] Cho S, Lee J, Park H, et al. Detection of arousals in patients with respiratory sleep disorders using a single channel EEG. In: 27th annual conference of the IEEE engineering in medicine and biology society, Shanghai, China; 2005.

[7] Agarwal R. Automatic detection of micro-arousals. In: 27th annual conference of the IEEE engineering in medicine and biology society, shanghai, China; 2005.

[8] Pittman S, MacDonald M, Fogel R, et al. Assessment of automated scoring of polysomnographic recordings in a population with suspected sleep-disordered breathing. Sleep 2004;27(7):1394–403.

[9] Malinowska U, Durka P, Blinowska K, et al. Micro- and macrostructure of sleep EEG. IEEE Eng Med Biol Mag 2006;25(4):26–31.

[10] Sugi T, Kawana F, Nakamura M. Automatic EEG arousal detection for sleep apnea syndrome. Biomed Signal Process Control 2009;4(4):329–37.

[11] Shmiel O, Shmiel T, Daga Y, et al. Data mining techniques for detection of sleep arousals. J Neurosci Methods 2009;179:331–7.

[12] Alvarez-Estevez D, Moret-Bonillo V. Identification of electroencephalographic arousals in multichannel sleep recordings. IEEE (Inst Electr Electron Eng) Trans Biomed Eng 2011;58(1):54–63.

[13] Alvarez-Estevez D, Sánchez-Maroño N, Alonso-Betanzos A, et al. Reducing dimensionality in a database of sleep EEG arousals. Expert Syst Appl 2011;38(6):7746–54.

[14] Alvarez-Estevez D. Diagnosis of the sleep apnea-hypopnea syndrome: a comprehensive approach through an intelligent system to support medical decision. 2012.

[15] Fernández-Varela I, Hernández-Pereira E, Alvarez-Estevez D, et al. Combining machine learning models for the automatic detection of EEG arousals. Neurocomputing 2017;268:100–8.

[16] Fernández-Varela I, Alvarez-Estevez D, Hernández-Pereira E, et al. A simple and robust method for the automatic scoring of EEG arousals in polysomnographic recordings. Comput Biol Med 2017;87:77–86.

[17] Quan S, Howard B, Iber C, et al. The sleep Heart Health study: design, rationale, and methods. Sleep 1997;20(12):1077–85.

[18] Kemp B, Olivan J. European data format 'plus' (EDF+), an EDF alike standard format for the exchange of physiological data. Clin Neurophysiol 2003;114: 1755–61.

[19] The National Sleep Research Resource, "The National Sleep Research Resource," [Online]. Available: http://sleepdata.org. [Accessed 2018].

[20] Dean D, Goldberger A, Mueller R, et al. Scaling up scientific discovery in sleep medicine: the national sleep research resource. Sleep 2016;39(5):1151–64.

[21] "Sleep Health Heart Study at NSRR," [Online]. Available: https://sleepdata.org/datasets/shhs/. [Accessed 2018].

[22] Redline S, Sanders M, Lind B, et al. Methods for obtaining and analyzing unattended polysomnography data for a multicenter study. Sleep Heart Health Res Gr 1998;21(7):759–67.

[23] Rechtschaffen A, Kales A. A manual of standardized terminology, techniques and scoring system of sleep stages in human subjects, Los Angeles. 1968.

[24] Iber C, Ancoli-Israel S, Chesson A, et al. The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specifications, Westchester, IL. 2007.

[25] The Atlas Task Force of the American Sleep Disorders Association. EEG arousals: scoring rules and examples. Sleep 1992;15(2):173–84.

[26] Case Western Reserve University. Sleep heart health study: reading center manual of operations. Cleveland, Ohio: Case Western Reserve University; 2002.

[27] "Sleep Heart Health Study. Montage and sampling rate information SHHS2," [Online]. Available: https://sleepdata.org/datasets/shhs/pages/5-montage-and-sampling-rate-information-shhs2.md. [Accessed 2018].

[28] Alvarez-Estevez D, van Velzen I, Ottolini-Capellen T, et al. Derivation and modeling of two new features for the characterization of rapid and slow eye movements in electrooculographic sleep recordings. Biomed Signal Process Control 2017;35:87–99.

[29] Fernández-Varela I, Alvarez-Estevez D. GitHub - arousals-detection [Online]. Available: https://github.com/bigsasi/arousals-detection; 2018.

[30] "SHHS dataset files at NSRR," [Online]. Available: https://sleepdata.org/datasets/shhs/files/datasets. [Accessed 2018].

[31] Shrout P, Fleiss J. Intraclass correlations: uses in assessing rater reliability. Psychol Bull 1979;86(2):420–8.

[32] McGraw K, Wong S. Forming inferences about some Intraclass correlation coefficients. Psychol Methods 1996;1(1):30–46.

[33] Salarian A. Intraclass Correlation Coefficient (ICC) at MathWorks File Exchange [Online]. Available: https://nl.mathworks.com/matlabcentral/fileexchange/22099-intraclass-correlation-coefficient-icc; 2017. Accessed 2018.

[34] Nakagawa S, Schielzeth H. Repeatability for Gaussian and non-Gaussian data: a practical guide for biologists. Biol Rev 2010;85:935–56.

[35] "rptR: Repeatability estimation for Gaussian and non-Gaussian data," [Online]. Available: http://rptr.r-forge.r-project.org/. [Accessed 2018].

[36] Kraemer H, Korner A. Statistical alternatives in assessing reliability, consistency, and individual differences for quantitative measures: application to behavioral measures of neonates. Psychol Bull 1976;83(5):914–21.

[37] Hallgren K. Computing inter-rater reliability for observational data: an overview and tutorial. Tutorials Quantitative Methods Psychol 2012;8(1):23–34.

[38] Drinnan M, Murray A, Griffiths C, et al. Interobserver variability in recognizing arousal in respiratory sleep disorders. Am J Respir Crit Care Med 1998;158(2): 358–62.

[39] Loredo J, Clausen J, Ancoli-Israel S, et al. Night-to-night arousal variability and interscorer reliability of arousal measurements. Sleep 1999;22(7):916–20.

[40] Ruehland W, Churchward T, Schachter L, et al. Polysomnography using abbreviated signal montages: impact on sleep and cortical arousal scoring. Sleep Med 2015;16:173–80.

[41] Wong T, Galster P, Lau T, et al. Reliability of scoring arousals in normal children and children with obstructive sleep apnea syndrome. Sleep 2004;27(6): 1139–45.

[42] Whitney C, Gottlieb D, Redline S, et al. Reliability of scoring respiratory disturbance indices and sleep staging. Sleep 1998;21(7):749–57. 21.

[43] Smurra M, Dury M, Aubert G, et al. Sleep fragmentation: comparison of two definitions of short arousals during sleep in OSAS patients. Eur Respir J 2001;7:723–7.

Chapter 3

Detection of Sleep Spindles

3.1 A Comparison of Performance of Sleep Spindle Classification Methods Using Wavelets

- **Title:** A Comparison of Performance of Sleep Spindle Classification Methods Using Wavelets
- **Authors:** Elena Hernández-Pereira, Isaac Fernández-Varela, Vicente Moret-Bonillo
- **Conference:** Innovation in Medicine and Healthcare 2016
- **Volume:** 60
- **Year:** 2016
- **Pages:** 61-70
- **DOI:** 10.1007/978-3-319-39687-3_6
- **ISBN:** 9783319396866
- **ISSN:** 21903026
- **Available in:** `http://link.springer.com/10.1007/978-3-319-39687-3_6`

A Comparison of Performance of Sleep Spindle Classification Methods Using Wavelets

Elena Hernandez-Pereira, Isaac Fernandez-Varela and Vicente Moret-Bonillo

Abstract Sleep spindles are transient waveforms and one of the key features that contributes to sleep stages assessment. Due to the large number of sleep spindles appearing on an overnight sleep, automating the detection of this waveforms is desirable. This paper presents a comparative study over the sleep spindle classification task involving the discrete wavelet decomposition of the EEG signal, and seven different classification algorithms. The main goal was to find a classifier that achieves the best performance. The results reported that Random Forest stands out over the rest of models, achieving an accuracy value of 94.08 ± 2.8 and 94.08 ± 2.4 % with the symlet and biorthogonal wavelet families.

Keywords Sleep spindles · Wavelets · Machine learning

1 Introduction

According to the current AASM definition [3], the Sleep Spindle (SS) is a "train of distinct waves with frequency 11–16 Hz (most commonly 12–14 Hz) with a duration greater or equal to 0.5 s, usually maximal in amplitude in the central derivations". The sleep spindle waves are characterized by progressively increasing then gradually decreasing amplitude, that may be present in low voltage background Electroencephalogram (EEG), superimposed to delta activity, or temporally locked to a vertex sharp wave and to a K complex [16]. Spindles are one of the key features that contributes to sleep stages assessment, specifically is one of the hallmarks of Non-Rapid

E. Hernandez-Pereira (✉) · I. Fernandez-Varela · V. Moret-Bonillo
Faculty of Informatics, Department of Computer Science,
University of A Coruña, Campus de Elviña s/n, 15071 A Coruña, Spain
e-mail: elena.hernandez@udc.es

I. Fernandez-Varela
e-mail: isaac.fvarela@udc.es

V. Moret-Bonillo
e-mail: vicente.moret@udc.es

Y.-W. Chen et al. (eds.), *Innovation in Medicine and Healthcare 2016*,
Smart Innovation, Systems and Technologies 60,
DOI 10.1007/978-3-319-39687-3_6

Eye Movement (NREM) stage 2 sleep, both in adults and children. Unfortunately, their visual identification is very time-consuming (there are typically hundreds of sleep spindles in a full night recording), and they are borderline in frequency or duration, or superimposed on other waveforms. Moreover, there are varying definitions of sleep spindle in the literature, making the criterion used for spindle scoring inconsistent across studies. Another limitation is that interscorer reliability for visual identification suggests a variability between scorers possibly due to subjectivity or expertise of the scorer [27]. Thus, automated sleep spindle detectors have been developed to reduce the workload of experts and eliminate the subjectivity.

The earliest sleep spindle detectors were dependent upon hardware [13, 20]. After these detection systems several software solutions have been attempted. Two principal approaches become accepted: those using band-pass filtering and amplitude detection, and those applying feature extraction follow by decision-making for classification. The first approach, followed in [10, 29], suffers from the interscorer variability and that is one of the reasons for the second approach to be a noted research line. Concerning algorithms based on features extraction followed by classification, the Short Time Fourier Transform (STFT) is a suitable tool to identify the frequency content of the sleep spindles. In [17], Gorur used STFT coefficients as inputs of a classifier. An agreement rate of 88.7 and 95.4 % were obtained with a multilayer perceptron (MLP) and a support vector machine (SVM) respectively. Another method used for features extraction is adaptive autoregressive modelling (AAR). In [1] the AAR coefficients were used as inputs for different classifiers: a discrete perceptron, a MLP and a SVM. The results obtained were compared in terms of sensitivity, achieving values of 99.2 %, 89.1 % and 94.6 % respectively. In recent years advanced time-frequency analysis tools like wavelets have been applied to the sleep EEG to derive improved feature vectors for sleep spindles. Ahmed et al. [2] proposed a automatic detector based upon the Teager Energy Operator (TEO) and Wavelet Packet Energy Ratio, and achieved an accuracy of 93.9 %. In [11] a multi-resolution decomposition technique based on wavelets and STFT, is developed to detect sleep spindles. After the detection, TEO is applied to determine spindle duration. By this approach, an overall sensitivity and specificity of 96.17 and 95.54 % were achieved. TEO is employed too in [19] where this operator isolated candidate spindle zones on sleep EEG and spectral edge frequency confirmed its presence. The algorithm used a normalized threshold and did not require patient-specific adjustments. It achieved 80 % and 97.6 % values of sensitivity and specificity respectively. Günes et al. [18] proposed a hybrid method based on time and frequency domain features. Welch spectral analysis has been used for the extraction of frequency domain features and a MLP for classification. The obtained classification accuracies for three feature sets (only time domain, only frequency domain and both frequency and time domain features) were 100, 56.86 and 93.84 %. In [24], an algorithm that models the amplitude frequency spindle distribution with a bivariate normal distribution is proposed. Spindle detection is not directly based on amplitude and frequency thresholds, but instead on a spindle distribution model that is automatically adapted to each individual subject. Authors concluded that normal modelling enhanced performance and improved spindle detection quality.

This work studies the capabilities of several machine learning techniques to classify sleep spindles. The feature extraction is accomplished using a discrete wavelet decomposition applied to the raw samples of the EEG signal segments. The paper is structured as follows: Sect. 2 proposes the research methodology, Sect. 3 describes the experimental procedure used in the research, Sect. 4 presents the results obtained and finally, the conclusions are presented in Sect. 5.

2 Research Methodology

The main objective of this work is to obtain a method that achieves the best accuracy results in the sleep spindle classification task. Over the EEG signal from several sleep recordings, a set of isolated waveforms was obtained. Using these patterns, the coefficients of a discrete wavelet decomposition were used as inputs for several classifiers.

2.1 Data Set

Patient data was gathered from the Sleep Laboratory of the André Vésale Hospital in Belgium. It consists of eight whole-nights recordings coming from patients—4 men and 4 women aged between 31 and 53—with different pathologies. Two EOG channels, three EEG channels and one submental EMG channel were recorded. The sampling frequency was 200 Hz for six records of the complete data set, 100 and 50 Hz for the two remaining ones. A segment of 30 min was extracted from each night from the central EEG channel for spindles scoring. No effort was made to select good spindle epochs or noise free epochs, in order to reflect reality as well as possible. These segments were given to a medical expert for sleep spindle scoring. The total number of identified spindles was 289 [10].

2.2 Feature Extraction

The wavelet transform is an efficient tool for decomposing a signal into a fundamental function set and obtaining sub-band localization. Figure 1 depicts the wavelet decomposition tree.

In the first step, a high pass filter $g(n)$ and a low pass filter $h(n)$ are applied to the original signal $x(n)$. After the filtering process, half of the samples at high frequency are discarded according to Nyquist Criteria. This operation is performed recursively for every remaining sample and the desired frequency intervals are obtained. We can mathematically express this procedure as follows:

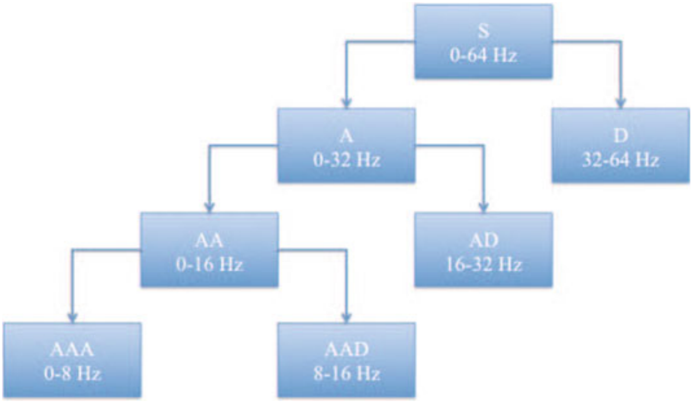


Fig. 1 Wavelet decomposition tree

$$Y_{high}[k] = \sum x[n]\dot{g}[2k - n] \tag{1}$$

$$Y_{low}[k] = \sum x[n]\dot{h}[2k - n] \tag{2}$$

where $Y_{high}[k]$ and $Y_{low}[k]$ are the outputs of the high pass (D) and low pass (A) filters, respectively.

The discrete wavelet transformation [26] provides a decomposition of a given signal into a set of approximation (a_i) and detail (d_i) coefficients of level i. The decomposition process can be iterated, with successive approximations being decomposed in turn, so that a signal is broken down into many lower-resolution components. Thus, in this case the samples of the EEG signal, are processed to obtain a level-1 transformation (a_1 and d_1 coefficients). Subsequently, each set of a_i coefficients is decomposed into a set of approximation a_{i+1} and detail d_{i+1} coefficients. Also, to obtain this decomposition some different types of wavelets functions can be used. The level-detail was determined taking into account the sample rate of the EEG signal and the wavelet families were chosen after performing some other experiments and discarding several wavelet families, specifically, the Symlet, Haar, Daubechies, Coiflets, Biorthogonal, and the discrete approach of the Meyer wavelet [9].

2.3 Classification

In this section, we provide an overview of the methods used in the research for sleep spindle classification. Several approaches were considered, two lineal models—a one-layer feedforward neural network and a proximal support vector machine—, and five non linear ones—a multilayer feedforward neural network, a classification tree, a Random Forest, a Support Vector Machine and a Naive Bayes classifier—.

- One-layer Feedforward Neural Network, One-lay FNN
  The one-layer feedforward neural network (FNN) is a single-layer FNN without hidden layers. This is a linear classification system that was trained using the supervised learning method proposed in [8]. The contribution of this learning method is that it is based on the use of an alternative cost function that measures the errors *before* the nonlinear activation functions instead of *after* them, as is normally the case. An important consequence of this formulation is that the solution can be obtained directly using a system of linear equations due to the fact that the new cost function is convex [14]. So, the method avoids local minima, and a very good approximation to the global minimum of the error function is obtained.
- Multilayer Feedforward Neural Network, FNN
  The multilayer feedforward neural network is one of the most commonly used neural network classification algorithms [4]. The architecture used for the classifier consisted of a three layer feed-forward neural network: two hidden and one output layer. The optimal number of hidden neurons for this problem was empirically obtained.
- Classification Trees, Class. Tree
  Classification trees are used to predict membership of cases or objects in the classes of a categorical dependent variable from their measurements on one or more predictor variables. In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels [5]. Each internal (non-leaf) node of the tree is labelled with an input feature. The arcs coming from a node labelled with a feature are labelled with each of the possible values of the feature. Each leaf of the tree is labelled with a class or a probability distribution over the classes. A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner. The recursion is completed when the subset at a node has all the same value of the target variable, or when splitting no longer adds value to the predictions. This process of top-down induction of decision trees is by far the most common strategy for learning decision trees from data [25].
- Random Forests, RF
  Random Forests [7] are an ensemble learning method for classification that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes. For an ensemble of decision trees for a multiclass classification function, one of the general methods is Bagging. This method is the simpler, more robust and more highly parallel technique. In the Bagging version used, a fixed-sized fraction of the training data is employed to construct each classifier in the ensemble. The Bagging method simply produces an ensemble of N decision trees constructed from N random subsets of the training data, where each subset is of the fixed-size mentioned in the previous sentence. With Bagging, the original method from the literature [6] of choosing a subset of points from a complete training set of N points was to choose a *bootstrap* sample [12]. Simply put, this means randomly choosing N points with equal probability from the set with replacement, so that some points may be chosen more than once or not at all.

To compute prediction of an ensemble of trees for unseen data, the Random Forest model takes an average of predictions from individual trees. To estimate the prediction error of the bagged ensemble, predictions for each tree are computed on its out-of-bag observations, are averaged over the entire ensemble for each observation and then the predicted out-of-bag response is compared with the true value at this observation.

- Support Vector Machine, SVM
  A Support Vector Machine is a supervised classification technique that works by nonlinearly projecting the training data in the input space to a feature space of higher (infinite) dimension by the use of a kernel function. This results in a linearly separable data set by a linear classifier. In many instances, classification in high dimension feature spaces results in overfitting in the input space; however, in SVMs, overfitting is controlled through the principle of structural risk minimization [28]. The empirical risk of misclassification is minimized by maximizing the margin between the data points and the decision boundary [21].
- Naive Bayes, NB
  Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. This assumption dramatically reduces the number of parameters that must be estimated to learn the classifier. Naive Bayes is a widely used learning algorithm, for both discrete and continuous inputs. The Naive Bayes Classifier technique is particularly suited when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods [23].
- Proximal Support Vector Machine, pSVM
  The proximal Support Vector Machine [15] is a method that classifies points assigning them to the closest of two parallel planes (in input or feature space) that are pushed as far apart as possible. The difference with a SVM is that this one classifies points by assigning them to one of two disjoint half-spaces. The pSVM leads to an extremely fast and simple algorithm by generating a linear or nonlinear classifier that merely requires the solution of a single system of linear equations.

3 Experimental Procedure

In order to characterize the performance of the system, the sensitivity and accuracy standard measures were used. The procedure presented has two stages: a feature extraction stage that establishes the main inputs for classification and the classification model itself. The initial step of the proposed methodology is the processing of the available EEG signals, obtaining isolated waveforms. The positive examples were identified by the medical expert and in order to achieve a balanced data set, the same number of negative examples were selected from the whole set of recordings. To get these negative examples, signal windows of 0.5 s (minimum sleep spindle duration) were randomly selected.

The sleep spindles contain low frequency components that are placed at 11–16 Hz band. Therefore, 8–16 Hz band covers all the information required for spindle detection. Thus, the computational cost and detection errors can be reduced by limiting the search to this sub-band. A discrete wavelet transformation was used as a pre-processing phase to reduce and fix the number of inputs of the classifier. Experimentally, for this work it was determined that the absolute value of the level-4-detail coefficients (d_4), the level-3-detail (d_3), and the level-2-detail (d_2) are the set of inputs that obtains the best sleep spindle classification results for the different sample rates (200 Hz, 100 Hz and 50 Hz respectively). Also, in order to make the decomposition, the symlet and biorthogonal wavelet families were used with a length of the filter equal to 10, 14 and 7 (symlet of order: O_5, sym5 and O_7, sym7; and biorthogonal of order $O_{1.5}$, bio1.5).

The number of coefficients supplied by the wavelet transformation depends on the number of samples of the supplied pattern. In this case, the minimum number of samples is 25 (as 0.5 s is the minimum duration of a sleep spindle and the lowest sample rate is 50 Hz), therefore, the number of corresponding wavelet coefficients is 13. As the number of inputs to the classification module must be fixed, the maximum number of coefficients that could be used is 13. For those patterns in which the duration is the minimum (0.5 s) all the coefficients (13) were used. For those other patterns with a duration greater than 0.5 s (for which more than 13 coefficients could be obtained) the first 13 coefficients were used as inputs to the classifier.

The experimental procedure is detailed as follows:

1. Extract the initial set of features to be used as inputs.
2. For each nonlinear classifier, establish its architecture. For the FNN a two hidden layer architecture with 10 and 8 units respectively was chosen. For the Random Forest, the number of trees chosen was 20 and for the SVM, the RBF kernel function was used.
3. Take the whole data set and generate 10-fold cross validation sets in order to better estimate the true error rate of each model. Eight folds are used to train the models, and the remainder ones to validate and test them respectively.
4. Train each model and obtain 10 performance measures over the validation sets and the test sets.
5. Select the best model in terms of accuracy.

The experiments performed in this work were executed using the software tool Matlab [22].

4 Results

In this section, the results obtained over the test set, after applying wavelet transformation and several classifiers are shown and compared in terms of accuracy and sensitivity. These results are yield against the standard reference, i.e. the medical expert scores. Tables 1 and 2 show the performance measures obtained.

Table 1 Sleep spindle classification results

	pSVM	One-lay. FNN	Class. Tree	RF	FNN	SVM	NB
sym5	86.20 ± 3.9	89.58 ± 3.9	91.83 ± 4.5	**94.08 ± 2.8**	87.89 ± 4.5	89.01 ± 4.3	93.66 ± 1.7
sym7	83.66 ± 6.8	85.35 ± 4.6	86.20 ± 3.2	**89.58 ± 3.2**	82.54 ± 6.6	86.90 ± 4.5	85.77 ± 5.2
bio1.5	86.20 ± 2.8	88.59 ± 2.7	93.38 ± 4.2	**94.08 ± 2.4**	86.48 ± 6.4	88.59 ± 3.9	93.66 ± 2.9

Mean test set accuracy (%) of a 10-fold cv. Best values marked in bold font

Table 2 Sleep spindle classification results

	pSVM	One-lay. FNN	Class. Tree	RF	FNN	SVM	NB
sym5	97.99 ± 1.9	96.01 ± 3.4	91.86 ± 6.8	95.78 ± 2.8	86.45 ± 6.1	92.67 ± 4.3	95.15 ± 1.5
sym7	95.04 ± 4.9	90.30 ± 3.8	85.47 ± 7.8	88.69 ± 6.5	82.16 ± 7.9	89.67 ± 4.5	89.63 ± 4.3
bio1.5	99.72 ± 0.8	98.89 ± 1.1	93.14 ± 4.3	95.81 ± 3.3	87.05 ± 7.9	92.08 ± 4.1	96.36 ± 3.6

Mean test set sensitivity (%) of a 10-fold cv

Among the linear models tested (pSVM and one-layer FNN), the one-layer FNN showed the best performance, achieving the highest accuracy for all the wavelet families used. For this classifier, the bio1.5 wavelet offers the best inputs. Over the nonlinear models, the Random Forest obtained the best results. These facts state no matter what wavelet family used. Nevertheless, the biorthogonal wavelet is the one that provides the best inputs for the classifier.

In terms of sensitivity, the linear models, pSVM and one-layer FNN, showed the highest values with the bio1.5 wavelet, but their accuracy values are not as good as expected. For the Random Forest model, the sensitivity values achieved were satisfactory for the sym5 and bio1.5 wavelets.

5 Conclusions

This paper presents a comparative study over the sleep spindle classification task involving the discrete wavelet decomposition of the EEG signal, and seven different classification algorithms. The main goal was to find a classifier that achieves the best accuracy results.

As a starting point, the extraction of isolated waveforms was carried out. Up to the authors knowledge, not many previous methods were proposed for sleep spindle classification that used the discrete wavelet decomposition as the feature extraction method. In this environment, several wavelets families were probed, being the symlet and biorthogonal families the ones that obtain the best results for the classifiers.

The results obtained were similar to those reported in the bibliography [2, 11] but a fair comparative study is not possible due to differences in both datasets and evaluation methods. In this work, from the classifier point of view, the results reported that Random Forest is the best option, achieving an accuracy value of 94.08 ± 2.8 and 94.08 ± 2.4 % with the symlet (order $O = 5$) and biorthogonal (order $O = 1.5$) wavelet families. For these models, the sensitivity values are similar (95.78 ± 2.8 and 95.81 ± 3.3 respectively). The results are encouraging and a deeper study will be done first in the negative examples extraction task. Different waveforms durations should be considered to make more difficult the classifier task instead of providing it with easy examples. Besides, we plan to test the use of ensembles of classifiers, trying to take advantage of the strengths of the different algorithms tested here and combine them in order to improve the classification accuracy. Finally, to confirm Random Forest best results, experiments over the entire signal length should be performed.

Acknowledgments This research was partially funded by the Xunta de Galicia (Grant code GRC 2014/035) and by the Spanish Ministerio de Economa y Competitividad, MINECO, under research project TIN2013-40686P both partially supported by the European Union ERDF.

References

1. Acir, N., Güzelis, C.: Automatic recognition of sleep spindles in EEG by using artificial neural networks. Expert Syst. Appl. **27**(3), 451–458 (2004)
2. Ahmed, B., Redissi, A., Tafreshi, R.: An automatic sleep spindle detector based on wavelets and the teager energy operator. In: Proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 2596–2599 (2009)
3. Berry, R.B., et al.: The AASM Manual for Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications. American Academy of Sleep Medicine, Darien, Illinois (2015)
4. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, New York (1995)
5. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Chapman & Hall, New York (1984)
6. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
7. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
8. Castillo, E., Fontenla-Romero, O., Alonso-Betanzos, A., Guijarro-Berdiñas, B.: A global optimum approach for one-layer neural networks. Neural Comput. **14**(6), 1429–1449 (2002)
9. Daubechies, I.: Ten lectures on wavelets. In: Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics (1992)
10. Devuyst, S., Dutoit, T., Stenuit, P., Kerkhofs, M.: Automatic sleep spindles detection. Overview and development of a standard proposal assessment method. In: 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC (2011)
11. Duman, F., Erdamar, A., Erogul, O., Telatar, Z., Yetkin, S.: Efficient sleep spindle detection algorithm with decision tree. Expert Syst. Appl. **36**(6), 9980–9985 (2009)
12. Efron, B.: Bootstrap methods: another look at the jackknife. Ann. Stat. **7**, 1–26 (1979)
13. Fish, D., Allen, P., Blackie, J.: A new method for the quantitative analysis of sleep spindles during continuous overnight eeg recordings. Electroencephalogr. Clin. Neurophysiol. **70**(3), 273–277 (1988)
14. Fontenla-Romero, O., Guijarro-Berdiñas, B., Pérez-Sánchez, B., Alonso-Betanzos, B.: A new convex objective function for the supervised learning of single-layer neural networks. Pattern Recognit. **43**(5), 1984–1992 (2010)

15. Fung, G., Mangasarian, O.: Proximal support vector machine classifiers. In: Provost, F., Srikant, R. et al. (eds.) Proceedings KDD-2001: Knowledge Discovery and Data Mining. pp. 77–86. San Francisco, CA, Asscociation for Computing Machinery, New York (2001)
16. Gennaro, L.D., Ferrara, M.: Sleep spindles: an overview. Sleep Med. Rev. **7**(5), 423–440 (2003)
17. Görür, D.: Automated Detection of Sleep Spindles. MSc thesis (2003)
18. Güneş, S., Dursun, M., Polat, K., Yosunkaya, C.: Sleep spindles recognition system based on time and frequency domain features. Expert Syst. Appl. **38**(3), 2455–2461 (2011)
19. Imtiaz, S.A., Saremi-Yarahmadi, S., Rodriguez-Villegas, E.: Automatic detection of sleep spindles using teager energy and spectral edge frequency. In: Biomedical Circuits and Systems Conference (BioCAS), 2013 IEEE, pp. 262–265 (2013)
20. Kumar, A., Hofman, W., Campbell, K.: An automatic spindle analysis and detection system based on the evaluation of human ratings of the spindle quality. Waking Sleep. 325–333 (1979)
21. Mashao, D.: Comparing SVM and GMM on parametric feature-sets. In: Proceedings of the 15th Annual Symposium of the Pattern Recognition Association of South Africa (2004)
22. MATLAB: version 8.4.0.150421 (R2014b). The MathWorks Inc., Natick, Massachusetts (2014)
23. Mitchell, T.: Machine Learning. McGraw Hill (1997)
24. Nonclercq, A., Urbain, C., Verheulpen, D., Decaestecker, C., Bogaert, P.V., Peigneux, P.: Sleep spindle detection through amplitude? Frequency normal modelling. J. Neurosci. Methods **214**(2), 192–203 (2013)
25. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**, 81–106 (1986)
26. Rao, R.M., Bopardikar, A.S.: Wavelet Transformations. Introduction to Theory and Applications (1998)
27. Ray, L.B., Fogel, S.M., Smith, C.T., Peters, K.R.: Validating an automated sleep spindle detection algorithm using an individualized approach. J. Sleep Res. **19**(2), 374–378 (2010)
28. Vapnik, V.: Statistical learning theory. Adaptive and learning systems for signal processing, communications, and control (1998)
29. Ventouras, E.M., Monoyiou, E.A., Ktonas, P.Y., Paparrigopoulos, T., Dikeos, D.G., Uzunoglu, N.K., Soldatos, C.R.: Sleep spindle detection using artificial neural networks trained with filtered time-domain EEG: a feasibility study. Comput. Methods Progr. Biomed. **78**(3), 191–207 (2005)

Chapter 4

Classification of Sleep Stages

4.1 Sleep staging with deep learning: a convolutional model

- **Title:** Sleep Staging with Deep Learning : A convolutional model
- **Authors:** Isaac Fernández-Varela, Dimitrios Athanasakis, Samuel Parsons, Elena Hernández-Pereira, Vicente Moret-Bonillo
- **Conference:** 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning
- **City:** Bruges (Belgium)
- **Dates:** 25-27 April
- **Year:** 2018
- **Pages:** 367-372
- **ISBN:** 978-287587047-6
- **Available in:**
`https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2018-59.pdf`

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

Sleep Staging with Deep Learning: A convolutional model

Isaac Fernández-Varela[1], Dimitrios Athanasakis[2], Samuel Parsons[3]
Elena Hernández-Pereira[1], and Vicente Moret-Bonillo[1] *

1- Universidade da Coruña - Departamento de Computación
Facultade de Informática, Campus de Elviña, A Coruña - Spain

2- Data Spartan
60 Ludgate Hill,London EC4M 7AW, United Kingdom

3- University College of London - Department of Computer Science
66-72 Gower Street, London WC1E 6EA, United Kingdom

Abstract. Sleep staging is a crucial task in the context of sleep studies that involves the analysis of multiple signals, thus being a very tedious and complex task. Even for a trained expert, it can take several hours to annotate the signals recorded from a patient's sleep during a single night. To solve this problem several automatic methods have been developed, although most of them rely on hand engineered features. To address the inner problems of this approach, in this work we explore the possibility of solving this problem with a deep learning network that can self-learn the relevant features from the signals. Particularly, we propose a convolutional network, obtaining higher performance than in previous methods, achieving an average precision of 0.91, recall of 0.90, and F-1 score of 0.90.

1 Introduction

Among the main tasks within the medical analysis of the sleep stands out the characterization of the sleep macro structure. Its final goal is the construction of the hypnogram, a graph that helps to interpretate the recorded electrical activities during a polysomnogram (PSG), showing the evolution of the different sleep stages through time.

The construction of the hypnogram was first proposed by Rechtschaffen and Kales (R&K) [1] in 1968 and only recently updated by the American Academy of Sleep Medicine (AASM) [2]. The method establishes a set of rules to assign labels (sleep stages) to time intervals typically lasting 30 s and called epochs. These sleep stages are: wakefulness (W), two stages for drowsy sleep (N1 & N2), one deep sleep (N3), and Rapid Eye Movement (REM).

Sleep staging is a tedious task, very time-consuming because it implies the analysis of multiple signals that record several hours (at least 6), thus there is a need to do it automatically. Several works address this problem with different approaches but they suffer from the problem of using hand engineered features.

*This research was partially funded by the Xunta de Galicia (Grant code GRC2014/035, and ED431G/01) partially supported by the European Union ERDF and from the Xunta de Galicia and the European Union Social Fund ESF.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

The latest review can be found in Penzel and Conradt [3]. Recent works already solve this problem avoiding hand engineered features [4, 5].

This works classifies sleep stages automatically, avoiding the use of hand engineered features using multiple signals at the same time. We also avoid the use of filters of methods to remove artifacts from the signals, feeding the network with the raw signals. The convolutional network that we are proposing is able to learn the relevant features from the signals to classify the sleep stages.

2 Materials

To develop and validate our proposal we have used real PSG recordings from the Sleep Heart Health Scoring database [6]. This database emerged from a multi-center cohort study to determine cardiovascular and other consequences of sleep-disordered breathing. Each recording includes off-line experts annotations following the R&K procedure [1]. The montage includes two EEG derivations (C4A2 and C4A1), rigth and left electrooculograms (EOG), bipolar submental electromyogram (EMG); and other signals which are not relevant to our problem. The EEG, EOG and EMG signals were recorded at 125 Hz and the EOG signals at 50 Hz. All the signals were filtered with a high pass filter set at 0.15 Hz during their acquisition.

A total of 240 recordings from different patients were randomly selected, using 180 for training, 20 for validation and 40 for testing. From each recording from train and validation sets and just to ease the model implementation, only 6 random hours were used, giving a total of 144,000 samples. The test set, which has a total of 49,794 samples, was used completely. No effort was done to select recordings with low noise ratio nor to discard segments with artifacts, as the model should be able to adapt to these situations.

In the train dataset 39.7% of the samples are classified as Awake, 38.3% as Drowsy Sleep, 9.6% as Deep Sleep, and 12.4% as REM. In the validation dataset the class distribution is: 42.0% for W, 37.3% for DS, 9.1% for N3, and 11.6% for REM. Finally, in the test dataset the distribution is: 42.7% for W, 37.3% for DS, 8.8% for N3, and 11.2% for REM.

3 Method

The goal of this work is to classify the different sleep stages of a PSG recording. As our first approximation, we simplify the problem using one label for drowsy sleep which includes both N1 and N2 stages. This was done in previous works [7] given that N1 is the stage for which expert classification presents the lowest inter-agreement [8] and also, the one with lower presence (only 3% of the epochs are classified as N1).

We solve this classification problem using a convolutional network[1]. A convolutional network is a deep feed-forward network that overcomes the limitations of multilayer perceptrons using a shared-weights architecture. The main reason

[1] Code and model are available in https://github.com/bigsasi/deepsleep

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

to use this network is its ability to learn the features that before were hand engineered.

To avoid biasing the model, we use as much data as it is possible to train it. Thus, we use the five available signals, although they are sampled at different rates. To overcome this problem, those signals sampled with a lower rate were padded with zeros. Then, a matrix with a row per signal was created. Obviously, this was done for each 30 seconds window, following the sleep stage definition. This way, each input to the network has a dimension of 3750 × 5. Although the input is bi-dimensional, our experiments were all done using 1D convolutional networks. With 1D convolutions we avoid imposing some artificial spatial structure between the different signals. Each signal was normalized to zero mean and unit standard deviation using train set as reference.

The convolutional model was selected using the validation set, trying to obtain the smallest network. From one layer models, we kept adding more layers until the performance did not improve. The performance of the model was defined as the average classification recall in the validation set. This experimentation led towards the model represented in Figure 1, which is composed of the following layers: two convolutional layers each with 128 kernels, one pool layer, another convolutional layer with 256 kernel, a max pool layer, and a final fully connected layer.

The filter size was fixed at 20 for every convolutional layer, with padding adjusted to maintain the input dimension. This value was selected after trying values from 3 (recommended value for convolutional networks used in artificial vision) to 65 (which would cover half a second of our signals). We observed that the performance improved with the filter size, but only up to 20, and decaying afterwards. The gradient optimizer was Adam [9] (with learning rate $3e - 4$) and the activation functions for all the convolutional layers *relu*, except for a final *softmax* function. We also added a dropout [10] of 0.5 in the final layer as regularization to avoid over-fitting. With this configuration the network had a total of 997,380 trainable parameters. Training was done with batches of size 32 and finished using early stopping over the validation loss with a patience value of 3.

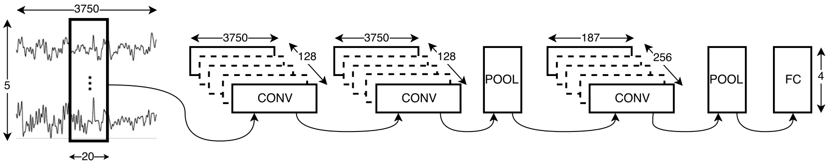


Fig. 1: Outline of the proposed convolutional network.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

4 Results

In order to validate the usefulness of the proposed approach an ensemble of 5 models was trained under the same dataset, but with random different 6 hours selection samples from each recording. To decide the final classification we use the mean value. Table 1 shows the global results obtained with our network and Table 2 the associated confusion matrix. The best precision value is obtained for the Awake class, achieving proximate values for the remaining classes. For the recall measure, it is worth mentioning the drop observed in the Deep Sleep stage.

	Precision	Recall	F-1 Score
Awake (W)	0.96	0.96	0.96
Drowsy Sleep (DS)	0.90	0.91	0.90
Deep Sleep (N3)	0.89	0.82	0.85
REM	0.89	0.90	0.90
Average	0.91	0.90	0.90

Table 1: Precision, recall and F-1 score for each sleep stage

	W	DS	N3	REM
W	20411	712	2	125
DS	741	16917	449	480
N3	1	796	3566	0
REM	152	392	0	5050

Table 2: Confusion matrix

Evaluating each recording individually, the distribution of the different measures is represented in Figure 2. This Figure confirms that the Awake class is the one with the best classification, with F-1 scores over 0.9, and and that the model struggles to classify Deep Sleep, with F-1 values falling to 0.4. Drowsy Sleep and REM classification present similar performance, with values over 0.8 for the F-1 score. Deep Sleep is the class with the highest deviation values, specially regarding the recall measure. Although precision is still greater than 0.8 for most of the records, recall values are worse, with measure values even as low as 0.3, which means that the model tends to underscore this class. Obviously, the fact is also reflected by the F-1 score, showing higher deviation than the other classes. In regards to outliers, the really low values (below 0.2) correspond to those recordings where the number of epochs classified (by the expert) as N3 is limited (lower than 2). Especially, the outliers represented as 0 correspond to a recording with no Deep Sleep epochs.

5 Conclusions

This work presents a method to classify sleep stages in PSG recordings using the common 30 seconds division called epoch. Our solution proposes the use of

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

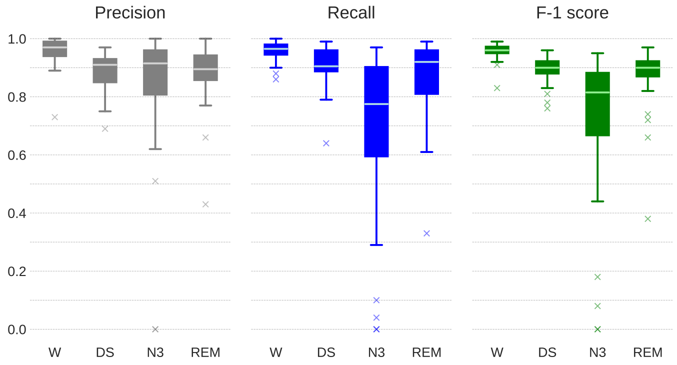
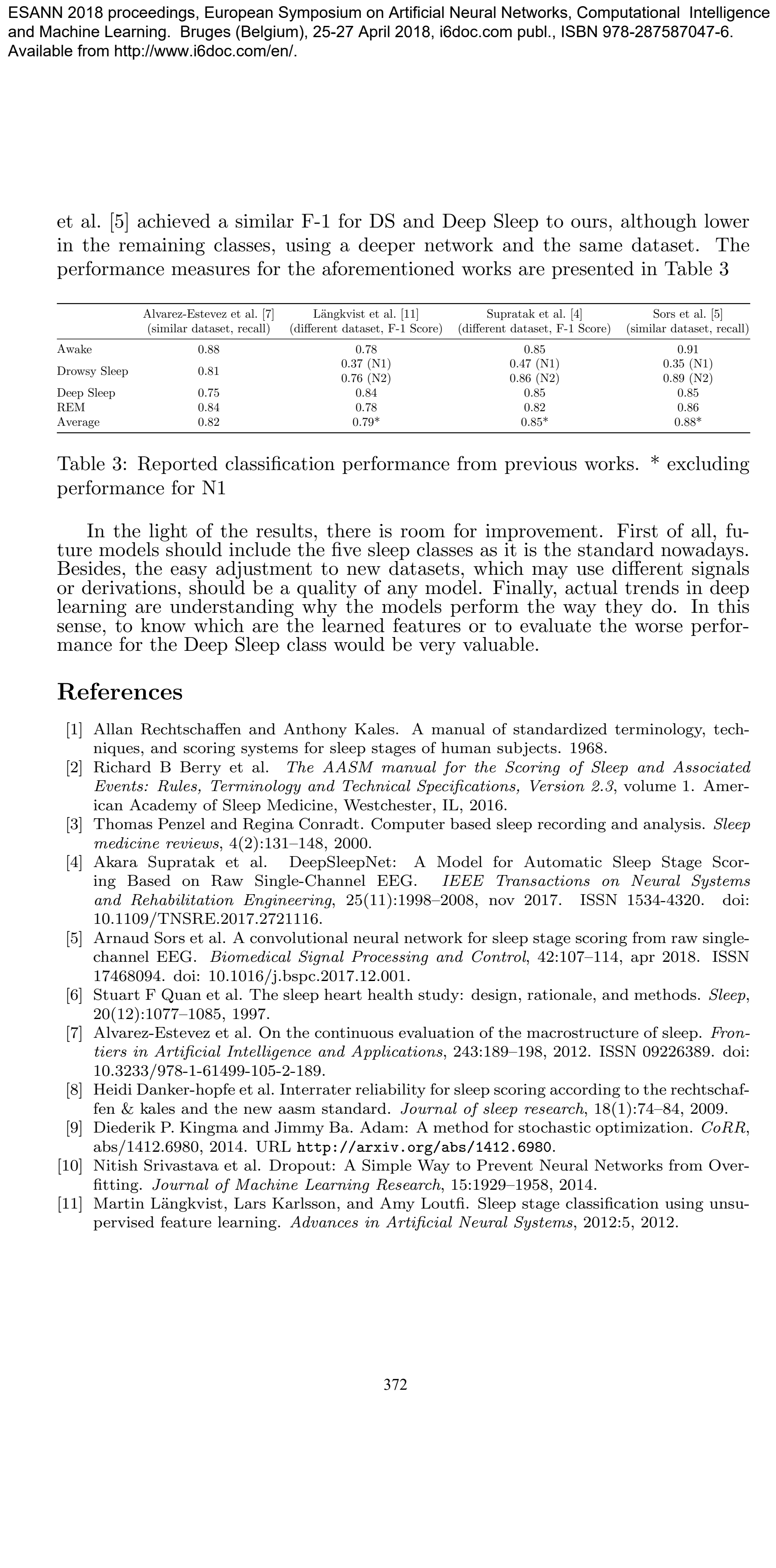


Fig. 2: Distribution of the performance measures for the individual recordings

a convolutional network that is feed with five available signals. The objective of this approach is to avoid human engineered features, which is how most works have solved this problem previously. The raw signals available in our dataset (2 EEG derivations, 1 EMG and 2 EOG) were directly used to feed the network. Thus, we also avoid the use of filters or signal preprocessing methods, apart from those applied within the hardware used to record the data.

Our model was selected using a validation dataset trying to achieve the highest recall with the fewer number of layers. Specifically, the architecture of our final model is composed of three convolutional layers with a pool layer after the second one and a fully connected layer at the end, with a total of 997,380 parameters trained. This yielded an average precision value of 0.91, a recall value of 0.90 and a F-1 score of 0.90.

It is difficult to compare our proposal against previous works due to the lack of benchmarks or clear methodology. Alvarez-Estevez et al. [7] presents a method using fuzzy logic after extracting hand engineered features from the signals. The method is validated on 26 recordings from the SHHS dataset, achieving an average recall value of 0.82, lower than the value obtained by the model described. For each sleep stage our method achieves between 7% (REM class) and 12% (DS class) higher recall. Längkvist et al. [11] avoid the use of hand engineered features with a deep belief network, although they remove noisy segments and select only those epochs with a clear label. Besides, they used a different dataset and classify 5 sleep stages. Assuming that their classification for drowsy sleep would be as good as it is for N2 stage (quite higher than for N1 stage), their average F1-score value is 0.79. Between classes, out method improved between 1% for the N3 class and 23% for the W class. Supratak et al. [4] uses multiples neural networks and a single EEG channel to classify with the Sleep-EDF dataset, achieving lower F-1 values compared to ours. Finally, Sors



et al. [5] achieved a similar F-1 for DS and Deep Sleep to ours, although lower in the remaining classes, using a deeper network and the same dataset. The performance measures for the aforementioned works are presented in Table 3

	Alvarez-Estevez et al. [7] (similar dataset, recall)	Längkvist et al. [11] (different dataset, F-1 Score)	Supratak et al. [4] (different dataset, F-1 Score)	Sors et al. [5] (similar dataset, recall)
Awake	0.88	0.78	0.85	0.91
Drowsy Sleep	0.81	0.37 (N1) 0.76 (N2)	0.47 (N1) 0.86 (N2)	0.35 (N1) 0.89 (N2)
Deep Sleep	0.75	0.84	0.85	0.85
REM	0.84	0.78	0.82	0.86
Average	0.82	0.79*	0.85*	0.88*

Table 3: Reported classification performance from previous works. * excluding performance for N1

In the light of the results, there is room for improvement. First of all, future models should include the five sleep classes as it is the standard nowadays. Besides, the easy adjustment to new datasets, which may use different signals or derivations, should be a quality of any model. Finally, actual trends in deep learning are understanding why the models perform the way they do. In this sense, to know which are the learned features or to evaluate the worse performance for the Deep Sleep class would be very valuable.

References

[1] Allan Rechtschaffen and Anthony Kales. A manual of standardized terminology, techniques, and scoring systems for sleep stages of human subjects. 1968.

[2] Richard B Berry et al. *The AASM manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications, Version 2.3*, volume 1. American Academy of Sleep Medicine, Westchester, IL, 2016.

[3] Thomas Penzel and Regina Conradt. Computer based sleep recording and analysis. *Sleep medicine reviews*, 4(2):131–148, 2000.

[4] Akara Supratak et al. DeepSleepNet: A Model for Automatic Sleep Stage Scoring Based on Raw Single-Channel EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(11):1998–2008, nov 2017. ISSN 1534-4320. doi: 10.1109/TNSRE.2017.2721116.

[5] Arnaud Sors et al. A convolutional neural network for sleep stage scoring from raw single-channel EEG. *Biomedical Signal Processing and Control*, 42:107–114, apr 2018. ISSN 17468094. doi: 10.1016/j.bspc.2017.12.001.

[6] Stuart F Quan et al. The sleep heart health study: design, rationale, and methods. *Sleep*, 20(12):1077–1085, 1997.

[7] Alvarez-Estevez et al. On the continuous evaluation of the macrostructure of sleep. *Frontiers in Artificial Intelligence and Applications*, 243:189–198, 2012. ISSN 09226389. doi: 10.3233/978-1-61499-105-2-189.

[8] Heidi Danker-hopfe et al. Interrater reliability for sleep scoring according to the rechtschaffen & kales and the new aasm standard. *Journal of sleep research*, 18(1):74–84, 2009.

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL `http://arxiv.org/abs/1412.6980`.

[10] Nitish Srivastava et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[11] Martin Längkvist, Lars Karlsson, and Amy Loutfi. Sleep stage classification using unsupervised feature learning. *Advances in Artificial Neural Systems*, 2012:5, 2012.

4.2 A convolutional network for the classification of sleep stages

- **Title:** A Convolutional Network for the Classification of Sleep Stages
- **Authors:** Isaac Fernández-Varela, Elena Hernández-Pereira, Diego Alvarez-Estevez, Vicente Moret-Bonillo
- **Note:** translation of the work *Una red convolucional para la clasificación de etapas de sueño*, CAEPIA 2018, I Workshop en Deep Learning. Granada 23-26 Octubre 2018. `https://doi.org/10.3390/proceedings2181174`
- **Available in:**
  `https://arxiv.org/abs/1902.05748`

A Convolutional Network for Sleep Stages Classification

Isaac Fernández-Varela*, Elena Hernández-Pereira*, Diego Alvarez-Estevez† and Vicente Moret-Bonillo*
*CITIC
Universidade da Coruña
A Coruña, España
(isaac.fvarela, elena.hernandez, vicente.moret)@udc.es
†Sleep Center & Clinical Neurophysiology
Haaglanden Medisch Centrum
The Hague, The Netherlands
diego.alvarez@udc.es

***Abstract*—Sleep stages classification is a crucial task in the context of sleep studies. It involves the simultaneous analysis of multiple signals recorded during sleep. However, it is complex and tedious, and even the trained expert can spend several hours scoring a single night recording. Multiple automatic methods have tried to solve these problems in the past, most of them by classifying a feature vector that is engineered for a specific dataset. In this work, we avoid this bias using a deep learning model that learns relevant features without human intervention. Particularly, we propose an ensemble of 5 convolutional networks that achieves a kappa index of 0.83 when classifying a dataset of 500 sleep recordings.**

***Index Terms*—convolutional network, sleep stages, classification**

I. Introduction

Sleep disorders affect a major part of the population. As an example, 20% of the Spanish adults suffer insomnia, and between 12% and 15% daytime sleepiness [1, 2]. Good sleep is essential for a healthy life, and the adverse consequences of restless nights have been extensively reported [3]. To evaluate the sleep function, and to help the diagnosis of sleep disorders, it is important to know the sequence of sleep stages that the patient goes through the night.

The most common technique to monitor the sleep function is the polysomnogram (PSG), which involves recording of the patient's biosignals during sleep, including various pneumological, electrophisiological, and contextual information. This is an expensive test, uncomfortable for the patient, and for which interpretation of the results is difficult due to the complexity of the data involved. An usual way to summarize the sleep information contained in the PSG is the derivation of the hypnogram, an ordered representation of the sleep stages evolution.

The current gold standard for the building the hypnogram is the *American Academy of Sleep Medicine* (AASM) [4] guide for the identification of sleep stages and of their associated events (e.g. EEG arousals, limb movements, and cardiac or respiratory events). This guide identifies five sleep stages: Awake (W), Rapid Eye Movements (REM), and 3 non-REM phases (N1, N2, and N3). Correct identification of the sleep stages and construction of the hypnogram is of fundamental importance to achieve a good diagnosis, allowing the clinician to focus efforts in the therapy. Such a task implies the analysis of huge amounts of data and expert knowledge [5]. Moreover, even following the guidelines, inter-expert agreement usually remains below the 90%. For example, Stepnowsky et al. [6] studied the agreement between two experts finding kappa index values between 0.48 and 0.89. Similarly, Wang et al. [7] found values between 0.72 and 0.85. Furthermore, agreement is worse for some particular stages, usually being stage N1 the one with the highest disagreement.

All given, automatic methods for sleep stages classification are needed. Most of these methods follow a two step approach. First, feature extraction takes place, usually with features hand tailored for a specific dataset. Then, feature vectors are built to train a classifier and predict the sleep stages. While some authors have used a single signal channel as reference (usually the EEG), other approaches have extracted features using several channels, building input vectors of various elements. At this respect usually features from the electrooculogram (EOG) or electromiogram (EMG) are added to those of the EEG, as recommended by the AASM guidelines. Often features are extracted either from to the time or from the frequency domain.

Among the methods following this 2-step approach we find: Fraiwan et al. [8] use a random forest to classify features both from the time-frequency domain and Renyi's entropy; Liang et al. [9] measure entropy with different scales obtaining autoregresive features which classify using a linear discriminant; Hassan and Bhuiyan [10], apply wavelet transformations for feature extraction and use a random forest technique for the classification step. Sharma et al. [11], compare several classifiers for iterative filters analysing a single EEG channel; Koley and Dey [12], train a support vector machine (SVM) with frequency, time and non-linear features extracted from a single EEG channel; Lajnef et al. [13], base their approach on multiple signals building a decision tree upon several SVMs;

* This research was funded by the Xunta de Galicia (ED431G/01) and the European Union (ERDF).

Huang et al. [14], study power spectral density of 2 EEG channels classifying frequency features with a modified SVM; Finally, Günes et al. [15], also analyse power spectral density while classifying with a nearest neighbours algorithm.

The approach consisting in solving the sleep staging classification problem using handcrafted feature extraction induces biases due to the design of features based on one specific database. Thus, the aforementioned solutions usually do not generalize well, specially given the nature of PSG recordings, where variability effects are introduced due to several factors, including patient, hardware or scoring differences.

One alternative option to solve this problem is the use of methods than learn directly from the raw data, therefore avoiding the human bias. In this sense, deep learning represents a natural approach, as it demonstrated improvements against traditional methods in multiple general fields, including in particular, the medical diagnosis [16, 17].

Some works have already explored solutions with different deep learning models: Längkvist et al. [18], used deep belief networks learning a probabilistic representation of preprocessed signals from PSG inputs; Tsinalis et al. [19], still followed the 2 step approach, but with convolutional networks for classification. In other work, the same authors [20] relied on a stack of *sparse autoencoders*; Supratak et al. [21], performed classification from the raw signals with a bidirectional recurrent neural network; Biswal et al. [22], compared a recurrent network against different models, although all were trained with features instead of the raw signal; Finally, Sors et al. [23] also used a convolutional neural network using one single EEG channel as reference.

In this work we use deep learning to classify sleep stages with a convolutional neural network that learns the relevant features for each stage. Following the AASM guidelines we use multiple signals; namely, two EEG, one EMG, and two (left and right) EOG channels. Moreover, signals are filtered in the first place, to reduce noise and remove artifacts.

II. Materials

Design and analysis of the presented model was carried out using PSG recordings from real patients. These recordings belong to the Sleep Heart Health Study (SHHS) [24], a database offered by the Case Western University, originated from a cohort study involving multiple centers directed by the National Heart Lung and Blood Institute, with the goal of determining the cardiovascular consequences of respiratory related sleep disorders.

Each recording contains annotations for different events performed by clinical experts following the procedures described in [25]. All recordings were anonymized and blind scored. The montage for the signals acquisition included two EEG derivations (C4A2 and C4A1), left and right EOGs, chin EMG, and modified lead-II electrocardiogram (ECG). EEG, EOG, and EMG were sampled at 125 Hz whereas EOG were sampled at 50 Hz. All signals were filtered during acquisition with a high pass filter at 0.15 Hz.

From this database three different datasets were selected to train, validate and test our model. Training dataset included 400 recordings, validation 100, and test 500. The length of the training recordings is matched (limiting each to a total of 7 randomly selected hours) to facilitate the coding and the training of the algorithm. Finally, our training dataset contained 288.000 $30-s$ epoch samples, the validation dataset 119.121 and the test dataset 606.981. Recordings were selected randomly, including those with high levels of noise or artifacts.

The distribution for the different classes, both for the complete dataset as for each individual recording is shown in Table I. This table shows how unbalanced the datasets are, being W the most represented class (about 38% of the samples), although with a similar proportion to N2 (around 36%). On the contrary, class N1 is only represented in 3% of the classes It is also interesting to notice how some recordings do not contain samples for some of the classes, and how much the distribution differs between the recordings. For example, in the test dataset, whereas a particular recording contains a 7.10% of samples for class N2, another goes up to a 83.43%. Moreover, these are the two important problems when trying to develop an automatic sleep staging classifier: 1) the class unbalance and 2) the differences between individual recordings.

III. Methods

A. Signal filtering

Signals are preprocessed to reduce noise and remove common artifacts. Both operations are typically applied in previous works before feature extraction.

The first of the two filters used to reduce noise is a Notch filter centered at 60 Hz to remove mains interference. This filter is applied to those signals with a sampling rate higher than 60 Hz: EEG and EMG. The second one removes DC component and frequencies not related with muscular movements from the EMG, applying a high pass at 15 Hz.

Regarding artifacts, most of then happen during particular short time periods, making it difficult even their detection. However, ECG artifacts, caused by the heart beat interference, are common and constant through the whole signals. We can remove this kind of artifact with an adaptive filter. To do so, we first obtained the beat series following a standard QRS detection algorithm [26]. Then, we studied the signal quality to asses which intervals could be safely included in the construction of the adaptive filter. Finally, during the intervals with enough signal quality, we applied and updated the filter template to remove the artifacts. More information about this process can be found in Fernández-Varela et al. [27].

B. Convolutional network

Sleep stages classification is usually carried out with 30 s windows called epochs. Analyzing several features from each epoch, clinicians score the corresponding sleep stage.

A convolutional neural network [28] is a feedforward network solving the limitations of the multilayer perceptron with a weight sharing architecture. Basically, it applies a

TABLE I
Distribution of the different classes in the training, validation, and test datasets.

		W	N1	N2	N3	REM	Total
Training dataset	Total	187.513	17.283	172.451	44.454	62.168	483.869
	Proportion	38,75 %	3,57 %	35,64 %	9,19 %	12,85 %	100 %
	Min in single record	8,20 %	0,00 %	12,59 %	0,00 %	0,00 %	
	Max in single record	71,61 %	13,75 %	68,65 %	33,43 %	26,58 %	
Validation dataset	Total	43.742	3.963	43.510	12.900	15.006	119.121
	Proportion	36,72 %	3,33 %	36,53 %	10,83 %	12,60 %	100 %
	Min in single record	11,21 %	0,29 %	12,38 %	0,00 %	0,00 %	
	Max in single record	76,79 %	17,08 %	60,09 %	30,16 %	23,68 %	
Test dataset	Total	231.707	19.769	217.246	61.281	76.978	606.981
	Proportion	37,77 %	3,26 %	35,96 %	10,25 %	12,75 %	100 %
	Min in single dataset	7,75 %	0,00 %	7,10 %	0,00 %	0,00 %	
	Max in single dataset	76,53 %	16,93 %	83,43 %	43,82 %	31,11 %	

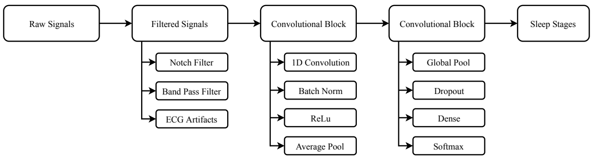


Fig. 1. Proposed convolutional neural network

convolution operation over the input, limiting the number of parameters. Thus, it allows the construction of deeper networks that are better at recognizing complex features. The proposed network is represented in Figure 1.

The input to the convolutional network is the set of signals (2 EEG channels, EMG, and both EOGs). Each input pattern corresponds to a 30 s epoch window. As the signals are sampled at different rates (aforementioned in Section II) we upsampled those with sampling rates lower than 125 Hz. We avoided downsampling to 50 Hz because it would mean loosing high frequencies in the EEG that should contain important information from a clinical perspective. Moreover, we also discarded padding because the approach cannot be easily generalized to other datasets with different sampling rates. This way, each input to the network is a matrix with a dimension of 3750×5. Each signal was normalized with mean 0 and deviation 1, using the mean and deviation obtained from all the respective signals in the training dataset. When we tried other normalizations with lower granularity, our training did not converge. The convolutional block shown in Figure 1 is a stack of four layers including a 1D convolution that preservers the input dimension (with padding), a batch normalization layer [29] to improve regularization, ReLu [30] activation, and an average pool that reduces dimension by a factor of 2. By using 1D convolution we avoided imposing a spatial structure between our signals that is unknown a priori. This stack was repeated n times, being n an hyperparamenter with a value selected during experimentation. All layers were configured with the same kernel size but the number of filters for layer i is twice the number of filters for layer $i-1$. The selection value of n, the kernel size and the number of filters for the first layers is explained in the following Section, together with the remaining hyperparameters.

The output of the last convolutional block, after adjusting dimensions with a global pooling and applying dropout, is used as input for a dense layer with a softmax activation. This layer returns the probability for each sleep stage given the initial input. As usual, the final predicted class is set to the output showing the highest probability.

To train the network we used Adam optimizer [31] and a batch size of 64. This batch size was limited by our hardware. The learning rate was configured whereas both betas are left with the default values. Training ends using early stopping by monitoring the validation loss with a patience of 10 epochs. To limit the impact of class unbalance, we used weighted cross entropy as the cost function, where weights were obtained using the training dataset.

C. Hyperparameter optimization

A good selection of hyperparameters can mean the success of a deep learning model. The difficulty when selecting the best hyperparameters is not only to achieve the best performance, but doing it while at the same time minimizing the cost, either the economical or the computational cost.

In this work we relied on a Tree-structured Parzen Estimator (TPE) that has shown better performance than other methods [32, 33]. TPE is a sequential models based optimization. This kind of methods builds models sequentially to approximate the performance of hyperparameters selection based on historical results, and then chooses new hyperparameters that are checked with the model. Particularly, TPE uses two distributions $P(x|y)$ and $P(y)$ where x represents the hyperparameters and y the expected performance. The expected improvement (EI) is optimized according to the following equation:

$$EI_{y^*}(x) = \int_{-\infty}^{y^*} (y^* - y) \frac{P(x|y)P(y)}{P(x)}$$

where y^* is a quantil γ of the observed values y such as $p(y < y^*) = \gamma$.

We used TPE to select the best values for the following hyperparameters related with the convolutional network: the number of convolutional blocks, kernel size for the 1D convolutions, and the number of filters for the first convolutional block. Moreover, there is also a relationship between the number of blocks and the number of initial filters. Given our hardware restrictions, we did not add blocks that would have more than 1024 filters. We also used TPE to select the learning rate. The distributions for the random values of each of these hyperparameters are summarized in Table II.

TABLE II
DISTRIBUTIONS FOR THE HYPERPARAMETERS

Hyperparameter	Distribution
Convolutional Blocks	Uniform between 1 and 10
Kernel Size	Uniform between 3 and 50
First Block Filters	Choice between 8, 16, 32 o 64
Learning Rate	Log-uniform between -10 and -1

To reduce the computational time for the hyperparameter selection we used a subset from the training set in order to train, validate, and test the different models. This subset contained 250 recordings where 20 were used for validation during training, and 50 to test each model. In total, we tried 50 different hyperparameter configurations, using the kappa index obtained with the test set as the criterion to select the best one.

D. Performance

The performance of the models was evaluated using the following metrics:

- **Precision,** the fraction between true positives and the predicted positives.
- **Sensitivity**, the fraction between true positives and the samples belonging to that class.
- ***F1 score***, harmonic mean between precision and sensitivity.
- **Kappa**, agreement measure between two classifiers that takes into account the chances of random agreement. Perfect agreement gets a value of 1, and by chance a value of 0.

IV. Results

Before focusing on the results achieved with the final model, performance of the different models evaluated during the hyperparameters search is shown in Figure 2. Data in the figure suggest a clear trend toward low learning rates to ensure convergence.

To improve the results obtained by a single model we used an ensemble. Thereby, several models classify the same input, and the final decision is taken using the majority vote. In this case, we selected the 5 best models obtained during the hyperparameter selection. Values for the hyperparameters for each of those models are shown in Table II.

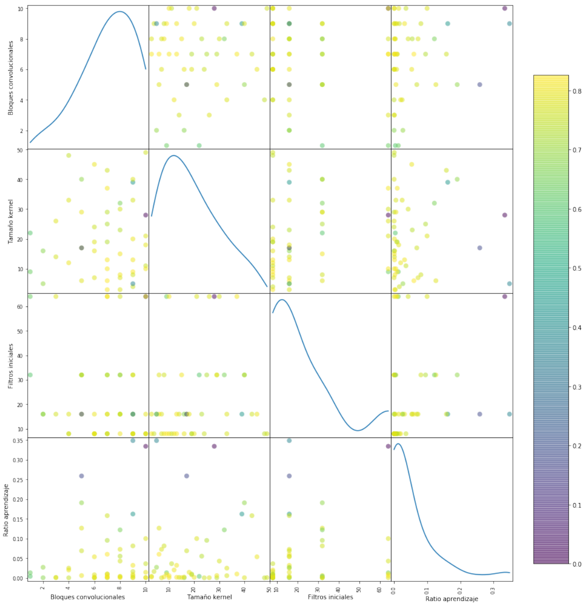


Fig. 2. Dispersion graph with the different configurations of hyperparameters. Each point color represents the kappa index for the model with the values for the hyperparameters represented in the axes. Diagonal represents the distribution for the values tried for a particular hyperparameter.

Results obtained with the ensemble using the test set are shown in Table IV. The best classification was achieved for class W, with values near to 0.95 for the precision, sensitivity and F1 score; then, classes N2, N3, and REM showed similar results, specially if we compare the F1 score, although sensitivity for N3 was lower (thus, precision was higher). Lastly, results regarding the the classification of class N1 were rather low, not even achieving a F1 score of 0.3. However, N1 is typically the most difficult class to predict, showing the highest disagreement also among trained experts.

The confusion matrix obtained with the ensemble is shown in Figure 3, where we can verify how most of the N1 samples are misclassified, specially towards class N2. Also, although in a smaller proportion, whenever there is a classification error it tends to be misclassifying as N2.

V. Discussion and Conclusions

In this work we present an ensemble of convolutional networks for the classification of sleep stages. Sleep staging is a time consuming task, nevertheless critical for a good diagnosis of sleep disorders. Most of the automatic methods reported so far are based on human engineered features, designed for a particular dataset. Thus, it is difficult to find a method that generalizes correctly to other datasets. To solve this problem we propose the use of a convolutional network that self learns the relevant features for the classification, avoiding human biases.

An important aspect for the success or failure of convolutional methods is the correct choice of the hyperparameters. In this paper, we experimented with 4 hyperparameters, op-

TABLE III
HYPERPARAMETERS FOR THE 5 MODELS WITH THE BEST KAPPA INDEX

Parameter	Model 1	Model 2	Model 3	Model 4	Model 5
Convolutional blocks	7	9	7	7	7
Kernel size	6	9	13	3	10
Initial filters	16	8	8	8	64
Learning rate	$5,99 \times 10^{-2}$	$9,00 \times 10^{-3}$	$1,45 \times 10^{-3}$	$1,91 \times 10^{-3}$	$5,49 \times 10^{-3}$

TABLE IV
PERFORMANCE MEASURES FOR THE CLASSIFICATION OF THE TEST DATASET USING THE ENSEMBLE WITH THE 5 SELECTED MODELS.

Stage	Precision	Sensitivity	*F1 score*
W	0,94	0,96	0,95
N1	0,39	0,21	0,27
N2	0,87	0,89	0,88
N3	0,92	0,77	0,84
REM	0,82	0,90	0,86
Average	0,78	0,75	0,76

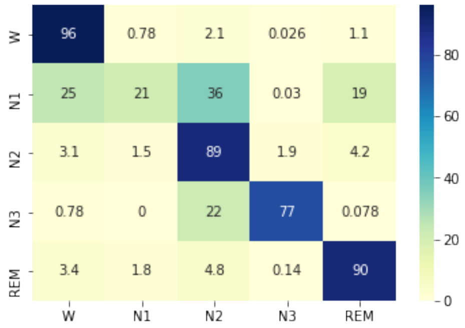


Fig. 3. Confusion matrix for the classification of the test dataset using the ensemble with the 5 selected models.

timizing their values with a tree-structured parzen estimator, trying 50 different configurations.

Our ensemble, built from the best 5 hyperparameters configurations, achieved an average precision, sensitivity, and F1 score of $0,78$, $0,75$ y $0,76$ respectively, with a kappa index value of 0.83. Although globally our results are acceptable, our solution has shown problems for the classification of class N1. Also, in the event of misclassification, a trend has been noticed towards class N2.

Comparison of our results against similar works is difficult given the lack of standardization, both as with regard to the chosen datasets, as well as in the procedures for the evaluation process. In Table V we show results from previous works, limiting to those that report values separately for each class. As it can be seen, our kappa index is the highest, although it is not the case for the F1 score. According to the F1 score, and apart from class W, some works are able to achieve better classification for the remaining classes. However, the values that we obtained are competitive, excluding class N1, although it is clear from all the results, that this is the most difficult class. Taking as reference the only work showing results with a similar dataset [23], our kappa index and F1 score for W class are higher, with similar values for N2, N3, and REM but lower for class N1.

Our results are promising and the chosen method should be easily adaptable to other datasets, specially if we can train the model for the different dataset. Moreover, training it with more than one dataset should improve generalization, avoiding biases for a single dataset.

To improve our result it is necessary to understand why and how the network is classifying. Also, it would be interesting to add memory to the model using recurrent networks, as the classification of some inputs, following the clinical definition, depends as well on the status of the neighbouring epochs.

REFERENCES

[1] M. M. Ohayon and T. Sagales, “Prevalence of insomnia and sleep characteristics in the general population of spain.” *Sleep medicine*, vol. 11, no. 10, pp. 1010–8, dec 2010.

[2] J. Marin *et al.*, “Prevalence of sleep apnoea syndrome in the spanish adult population,” *International Journal of Epidemiology*, vol. 26, no. 2, pp. 381–386, apr 1997.

[3] H. R. Colten and B. M. Altevogt, *Sleep Disorders and Sleep Deprivation*. Washington, D.C.: National Academies Press, sep 2006, vol. 6, no. 9.

[4] R. B. Berry *et al.*, “AASM Scoring Manual Updates for 2017 (Version 2.4).” *Journal of clinical sleep medicine : JCSM : official publication of the American Academy of Sleep Medicine*, vol. 13, no. 5, pp. 665–666, may 2017.

[5] Á. Fernández-Leal *et al.*, “A knowledge model for the development of a framework for hypnogram construction,” *Knowledge-Based Systems*, vol. 118, pp. 140–151, 2017.

[6] C. Stepnowsky *et al.*, “Scoring accuracy of automated sleep staging from a bipolar electroocular recording compared to manual scoring by multiple raters.” *Sleep medicine*, vol. 14, no. 11, pp. 1199–207, nov 2013.

[7] Y. Wang *et al.*, “Evaluation of an automated single-channel sleep staging algorithm.” *Nature and science of sleep*, vol. 7, pp. 101–11, 2015.

[8] L. Fraiwan *et al.*, “Automated sleep stage identification system based on time–frequency analysis of a single EEG channel and random forest classifier,” *Computer Methods and Programs in Biomedicine*, vol. 108, no. 1, pp. 10–19, oct 2012.

[9] J. Liang *et al.*, “Predicting seizures from electroencephalography recordings: A knowledge transfer strategy,” in *Proceedings - 2016 IEEE International Conference on Healthcare Informatics, ICHI 2016*. IEEE, oct 2016, pp. 184–191.

[10] A. R. Hassan and M. I. H. Bhuiyan, “A decision support system for automatic sleep staging from EEG signals using tunable Q-factor wavelet transform and spectral features,” *Journal of Neuroscience Methods*, vol. 271, pp. 107–118, sep 2016.

[11] R. Sharma, R. B. Pachori, and A. Upadhyay, “Automatic sleep stages classification based on iterative filtering of electroen-

TABLE V
COMPARISON AGAINST PREVIOUS WORKS.

Work	Database	Kappa	F1 score				
			W	N1	N2	N3	REM
Biswal et al. [22]	Massachusetts General Hospital, 1000 recordings	0,77	0,81	**0,70**	0,77	0,83	**0,92**
Längkvist et al. [18]	St Vicent's University Hospital, 25 recordings	0,63	0,73	0,44	0,65	**0,86**	0,80
Sors et al. [23]	SHHS, 1730 recordings	0,81	0,91	0,43	0,88	0,85	0,85
Supratak et al. [21]	MASS dataset, 62 recordings	0,80	0,87	0,60	**0,90**	0,82	0,89
Supratak et al. [21]	SleepEDF, 20 recordings	0,76	0,85	0,47	0,86	0,85	0,82
Tsinalis et al. [19]	SleepEDF, 39 recordings	0,71	0,72	0,47	0,85	0,84	0,81
Tsinalis et al. [20]	SleepEDF, 39 recordings	0,66	0,67	0,44	0,81	0,85	0,76
This work	SHHS, 500 recordings	**0,83**	**0,95**	0,27	0,88	0,84	0,86

cephalogram signals," *Neural Computing and Applications*, vol. 28, no. 10, pp. 2959–2978, oct 2017.

[12] B. Koley and D. Dey, "An ensemble system for automatic sleep stage classification using single channel EEG signal," *Computers in Biology and Medicine*, vol. 42, no. 12, pp. 1186–1195, 2012.

[13] T. Lajnef *et al.*, "Learning machines and sleeping brains: Automatic sleep stage classification using decision-tree multi-class support vector machines," *Journal of Neuroscience Methods*, vol. 250, pp. 94–105, jul 2015.

[14] C.-S. Huang *et al.*, "Knowledge-based identification of sleep stages based on two forehead electroencephalogram channels," *Frontiers in Neuroscience*, vol. 8, p. 263, sep 2014.

[15] S. Günes, K. Polat, and S. Yosunkaya, "Efficient sleep stage recognition system based on EEG signal using k-means clustering based feature weighting," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7922–7928, dec 2010.

[16] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, feb 2017.

[17] V. Gulshan *et al.*, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *JAMA*, vol. 316, no. 22, p. 2402, dec 2016.

[18] M. Längkvist *et al.*, "Sleep Stage Classification Using Unsupervised Feature Learning," *Advances in Artificial Neural Systems*, vol. 2012, pp. 1–9, 2012.

[19] O. Tsinalis *et al.*, "Automatic sleep stage scoring with single-channel eeg using convolutional neural networks," oct 2016.

[20] O. Tsinalis, P. M. Matthews, and Y. Guo, "Automatic sleep stage scoring using time-frequency analysis and stacked sparse autoencoders," *Annals of Biomedical Engineering*, vol. 44, no. 5, pp. 1587–1597, may 2016.

[21] A. Supratak *et al.*, "Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998–2008, nov 2017.

[22] S. Biswal *et al.*, "Sleepnet: Automated sleep staging system via deep learning," jul 2017.

[23] A. Sors *et al.*, "A convolutional neural network for sleep stage scoring from raw single-channel EEG," *Biomedical Signal Processing and Control*, vol. 42, pp. 107–114, apr 2018.

[24] S. F. Quan *et al.*, "The sleep heart health study: Design, rationale, and methods," *Sleep*, vol. 20, no. 12, pp. 1077–1085, dec 1997.

[25] Case Western Reserve University, "Sleep Heart Health Study: reading center manual of operations," Case Western Reserve University, Tech. Rep., 2002.

[26] V. Afonso *et al.*, "Ecg beat detection using filter banks," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 2, pp. 192–202, 1999.

[27] I. Fernández-Varela *et al.*, "A simple and robust method for the automatic scoring of EEG arousals in polysomnographic recordings," *Computers in Biology and Medicine*, vol. 87, pp. 77–86, aug 2017.

[28] Y. Le Cun *et al.*, "Handwritten digit recognition: applications of neural network chips and automatic learning," *IEEE Communications Magazine*, vol. 27, no. 11, pp. 41–46, nov 1989.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[30] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," *Proceedings of the 27th International Conference on Machine Learning*, no. 3, pp. 807–814, 2010.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," dec 2014.

[32] J. Bergstra *et al.*, "Algorithms for hyper-parameter optimization," in *NIPS*, 2011.

[33] J. Bergstra, D. Yamins, and D. D. Cox, "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms," in *Proc. of the 12th python in science conf*, 2013.

Chapter 5

Building an API for Sleep Medicine

5.1 A systematic approach to API usability

A systematic approach to API usability: Taxonomy-derived criteria and a case study

- **Title:** A systematic approach to API usability: Taxonomy-derived criteria and a case study
- **Authors:** Mosqueira-Rey, Eduardo, Alonso-Ríos, David, Moret-Bonillo, Vicente, Fernández-Varela, Isaac, Alvarez-Estevez, Diego
- **Journal:** Information and Software Technology
- **Editorial:** Elsevier
- **ISSN:** 09505849
- **Year:** 2018
- **Volume:** 97
- **Pages:** 46-63
- **DOI:** 10.1016/j.infsof.2017.12.010
- **Available in:**
`https://www.sciencedirect.com/science/article/pii/S0950584917302471`

Information and Software Technology 97 (2018) 46–63

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

A systematic approach to API usability: Taxonomy-derived criteria and a case study

Eduardo Mosqueira-Rey[a,*], David Alonso-Ríos[a], Vicente Moret-Bonillo[a], Isaac Fernández-Varela[a], Diego Álvarez-Estévez[b]

[a] *Department of Computer Science, University of A Coruña, Campus de Elviña, 15071 A Coruña, Spain*
[b] *Sleep Center & Clinical Neurophysiology – Haaglanden Medisch Centrum, Lijnbaan 32, 2512VA The Hague, The Netherlands*

ARTICLE INFO

Keywords:
Usability
Usability taxonomies
Usability studies
Application program interfaces
APIs
Sleep medicine

ABSTRACT

Context: The currently existing literature about Application Program Interface (API) usability is heterogeneous in terms of goals, scope, and audience; and its connection to accepted definitions of usability is rarely made explicit. The use of metrics to measure API usability is focused only on measurable characteristics excluding those usability aspects that are related to the subjectivity of human opinions.

Objective: Our objective is to build a comprehensive set of heuristics and guidelines for API usability that is a structured synthesis of the existing literature on API usability but which also covers other aspects that have been neglected so far. This set is explicitly connected with a usability model, something that allows us to check if we are addressing actual usability problems.

Method: Our approach is to follow a systematic approach based on a comprehensive model of usability and context-of-use. From this comprehensive model we derived the set of heuristics and guidelines that are used to carry out a heuristic evaluation with usability experts and a subjective analysis with users. The influence of the context of use, something that is normally ignored, is explicitly analyzed.

Results: Our heuristics and guidelines were integrated into a usability study of a sleep medicine API. In this study, we were able to identify several usability issues of the proposed API that are not explicitly addressed in the existing literature. The context of use helped us to identify those categories that were more relevant to consider in order to improve API usability.

Conclusion: The literature on API usability is very technically-minded and tends to neglect the subjective component of usability. We contribute to a more global and comprehensive view of the usability of APIs that is not contradictory but complementary with metrics. Our criteria ease the always necessary usability evaluation with human evaluators and users.

1. Introduction

An Application Program Interface (API) is a particular set of rules and specifications that software programs can follow in order to communicate with each other. It serves as an interface between programs and facilitates their interaction, just as a graphical user interface facilitates interaction between humans and computers.

Why are APIs so important in modern computer engineering? There is a famous quote attributed to Newton that says “If I have seen further, it is by standing on the shoulders of giants”. This metaphor expresses the idea that new discoveries are built on previous discoveries. In the same way, software construction nowadays is a task of building software on top of other software. For example, you can use the instruction set of the processor, the system calls of the operating system, the classes and methods of the core library of your programming language, or the additional libraries that you are using for different purposes (graphics, collections management, web services, etc.).

With the current popularity of web applications, it is common for them (e.g., Google Maps, YouTube, Facebook, or Twitter) to expose an API so programmers can interact with them and integrate them in their web pages or mobile applications. There is a website called *ProgrammableWeb*[1] that has an extensive directory of APIs available to web programmers. It has currently indexed over 15,000 APIs.

The problem with APIs is that so many of them are not well

* Corresponding author.
E-mail addresses: eduardo@udc.es (E. Mosqueira-Rey), dalonso@udc.es (D. Alonso-Ríos), civmoret@udc.es (V. Moret-Bonillo), isaac.fvarela@udc.es (I. Fernández-Varela), dalvareze@udc.es (D. Álvarez-Estévez).

[1] http://www.programmableweb.com.

https://doi.org/10.1016/j.infsof.2017.12.010
Received 17 March 2017; Received in revised form 8 December 2017; Accepted 23 December 2017
Available online 28 December 2017
0950-5849/

designed. Henning [27] stated that the prime reason is that it is very easy to create a bad API, but rather difficult to create a good one. Minor flaws are magnified when a considerable number of programmers are using the API. That means that we have to pay careful attention to usability issues related to API construction and use. However, how do we analyze the usability of an API?

Our approach is based on the idea that comprehensive models of usability can be used as the basis for studying the usability of APIs, just as they can be used to assess the usability of devices, graphical user interfaces, and so on. We also argue that usability heuristics and guidelines should be explicitly connected to the concept of usability itself, in order to address the full range of usability. This is why it is important to make explicit the connection between API heuristics (and guidelines) and the subjacent usability model.

This paper examines the existing literature on API usability and uses an expanded usability model to organize this information and identify deficiencies. As a result, we propose a comprehensive set of heuristics and guidelines for usability studies that is a structured synthesis of the existing literature but also covers other aspects that have been neglected to date.

These heuristics and guidelines are tested by means of a case study. The heuristics and guidelines are integrated into a usability study of an API for a decision support system for sleep medicine.

The paper is structured as follows. Section 2 is a background section in which we review the literature on API usability, introduce our expanded usability model and explain the field of application. Section 3 describes the methodology used in our study. Section 4 presents the results of our analysis, which consists in mapping the literature to the usability model and proposing new heuristics and guidelines. Section 5 describes the case study including the requirements identification, the heuristic evaluation and the subjective analysis. We finish with a discussion of the results, the conclusions, and plans for future work (Section 6).

2. Background

The best known definition of usability is probably the one in ISO 9241–11:1998 which defines usability as “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.” (ISO 9241–11, [29], p. 2). The simplicity of early usability models has been criticized by usability researchers and practitioners, who have approached this problem in different ways. For example, by producing extensive lists of heuristics and guidelines for usability studies. Another approach is to investigate and expand on the concept of usability itself. Researchers such as Bevan [12], Seffah et al. [47], Winter et al. [52], and Alonso-Ríos et al. [1] have created what Lewis [36] calls “expanded models of usability” . These expanded models refine the traditional usability models in the literature by integrating different models and breaking down attributes into subattributes.

2.1. Literature on API usability

One of the first publications that addresses the question of API usability was McLellan et al. [39]. They stated the basic idea that programmers are users too and that they need libraries that are just as easy to learn and use as the products they build from these libraries. They identify the following attributes to take into account when analyzing API usability: How easy the API is to learn, how efficiently the API can be used for specific tasks, how easy the API calls are to remember, what misconceptions or errors programmers make using the API, and how programmers perceive the API.

Other authors follow a methodological approach to API usability. For example Clarke [16, 17] based his work on the cognitive dimensions framework of Green and Petre [25]. This framework is defined by its authors as a “broad-brush evaluation technique for interactive devices and for non-interactive notations”. It sets out a small vocabulary of terms designed to capture the cognitively-relevant aspects of structure, and shows how they can be traded off against one another.

As a result, Green and Petre identified thirteen cognitive dimensions, namely: (1) *Abstraction Gradient*, (2) *Closeness of Mapping*, (3) *Consistency*, (4) *Diffuseness/Tenderness*, (5) *Error-proneness*, (6*) Hard Mental Operations*, (7) *Hidden Dependencies*, (8) *Premature Commitment*, (9*) Progressive Evaluation*, (10*) Role-expressiveness*, (11) *Secondary Notation and Escape from Formalism*, (12) *Viscosity: resistance to local change*, and (13) *Visibility and Juxtaposability*.

Clarke [16] adapted these dimensions to the API usability problem and identified 12 different dimensions or factors that individually and collectively have an impact on the way that developers work with an API and on the way that developers expect the API to work. These dimensions are: (1) *Abstraction level*, (2) *Learning style*, (3) *Working framework*, (4) *Work-step unit*, (5*) Progressive evaluation*, (6*) Premature commitment*, (7) *Penetrability*, (8) *API elaboration*, (9) *API Viscosity*, (10) *Consistency*, (11) *Role expressiveness*, and (12) *Domain correspondence*.

Clarke also proposes to follow a user-centered approach to designing usable APIs that uses scenarios to ensure that the API reflects the tasks that the users want to perform, rather than its implementation details. He also established developer profiles (opportunistic, pragmatic, and systematic) that would affect the way the usability of the API is analyzed.

However, the work of Clarke tends to be too abstract to be directly applicable by programmers [15], and it also leaves out important aspects of usability. We will see later in this paper that it does not take into account criteria related to documentation of the API or to the precision of the data types used. Bore and Bore [15] also proposed “going back to basic programming guidelines to derive a simple set of dimensions whose interpretation is clear”.

For example, there are publications that are more focused on programmers’ needs addressing specific problems or choices to be taken on API design and proposing solutions and courses of action. Some examples are the usability problems with the Factory Pattern ([21]) or the usability implications of requiring parameters in object constructors and the recommendation of using the create-set-call pattern, that is, objects can be created with default constructors and initialized later (Stylos & Clarke, [49]).

Other authors developed comprehensive sets of guidelines to help programmers develop usable APIs. Jacques [33] derived an API design checklist from general usability principles, allowing it to be used in inspections, walkthroughs and reviews. Henning [27] also proposed some guidelines and stated that “these guidelines do not guarantee success, but being aware of them during design makes it much more likely that the result will turn out to be usable”. Henning's guidelines are not especially detailed or comprehensive and tend to be very general.

Moreover, Zibran [53, 54] described a detailed set of 22 specific guidelines after an exhaustive study of existing literature. He stated that an API is usable if it has five characteristics: (1) *easy to learn*, (2) *easy to remember*, (3) *easy write client code*, (4) *easy to interpret client code*, and (5) *difficult to misuse*. However, he does not relate these characteristics to the proposed guidelines. Grill et al. [26] used the guidelines of Zibran and proposed a methodology on how to use and combine HCI methods with the goal to evaluate the usability of APIs. The methodology consists of three phases: a heuristic evaluation, a developer workshop, and interviews.

In addition, some of the most popular publications on API usability are textbooks from experienced API designers like Beck [9], Bloch [14], Martin [37], Tulach [51], or Cwalina and Abrams [18]. They published many guidelines that are more like “best practices” of programming rather than API usability guidelines but from which we can obtain useful recommendations for API design [13].

Other authors have approached the problem of API usability from the point of view of software metrics. A software metric is defined as a "function whose inputs are software data and whose output is a single numerical value that can be interpreted as the degree to which software possesses a given attribute that affects its quality" (IEEE [28], p. 3). One of the first well-known examples of software metric is the *cyclomatic complexity* defined by McCabe [38] used to indicate the complexity of a program by measuring the number of linearly independent paths through its source code. Metrics are used in many software-related areas like cost and effort estimation, quality and reliability models, security metrics, structural and complexity metrics, management metrics, etc. [22].

Bertoa et al. [11] presented a set of measures to assess the usability of software components. They based their study on the ISO/IEC 9126 [30] model that defines usability in terms of five sub-characteristics: (1) *Understandability*, (2) *Learnability*, (3) *Operability*, (4) *Attractiveness*, and (5) *Usability Compliance*. But the derived metrics from their work were rather abstract and it is not clear how they could be used in practice. Furthermore, the metrics are associated with two measurable concepts (*quality of the documentation* and *complexity of the design*) but are not associated with the usability sub-characteristic identified before.

Doucette [20] proposed 12 metrics that were strictly complexity metrics such as *number of classes, depth of inheritance hierarchy, average number of methods*, etc. Souza and Bentolila ([48], p. 299) proposed measuring API usability as a "function of its complexity, so that complex APIs are harder to use and maintain than APIs that are not complex". The problem with this approach is that API usability is more than API complexity and one cannot infer that an API is usable because it is not complex, and the other way around, a complex API can be perfectly usable if it is clear, consistent, well documented, etc.

Rama and Kak [45] proposed 9 new metrics that were created specifically to measure API usability. These metrics were structural measures such as: *the existence of too many methods with nearly identical names, not grouping conceptually similar API methods together, the poor quality of API documentation*, etc. The authors themselves state that these structural measures "do not constitute an exhaustive enumeration of all possible ways in which an API may exhibit structural defects" (p.82).

Scheller and Kühn [46] followed the works of Bertoa et al. [11] and Rama and Kak [45] and defined an API Concepts Framework, an extensible framework for measuring interface complexity. They defined 20 usability aspects and 30 potential measurable properties that can be used to measure these usability aspects. While the work disregards some aspects of usability due to the difficulty in measuring them by an automated framework (e.g., documentation, naming or abstraction level), the other aspects of usability are well measured.

Therefore, the common problem with the metric approach is that the works are somewhat associated with a usability model, but the metrics derived are not related with a usability characteristic or are associated with only a few of these usability characteristics (those that are easy to measure). Thus, those usability aspects that are related to subjectivity are not taken into account in these models. Scheller and Kühn [[46], p.146] recognized that "such kind of automated usability measure can never completely replace a thorough usability investigation with human evaluators or tests with users". Noticeably, Scheller and Kühn are one of the few authors that highlight the importance of taking into account not only the usability, but also the context of use when evaluating the usability of an API.

Finally, some authors like Daughtry et al. [19] published a good review of the state of the art of API usability studies. This state of the art was updated by Myers and Stylos [40] and some of these authors maintain a website (www.apiusability.org) as a repository for API usability related papers. Myers and Stylos [40] stated that "API designers should add usability as an explicit design and evaluation criterion so they do not create an unusable API inadvertently" and they promote the use of human-centered methods for improving API usability. As a means of evaluating an API design, they use the guidelines proposed by Nielsen [42] for performing a heuristic evaluation and map these heuristics to specific API guidelines.

Specifically, the Nielsen heuristic guidelines are as follows: (1) *Visibility of system status*, (2) *Match between system and the real world*, (3) *User control and freedom*, (4) *Consistency and standards*, (5) *Error prevention*, (6) *Recognition rather than recall*, (7) *Flexibility and efficiency of use*, (8) *Aesthetic and minimalist design*, (9) *Help users recognize, diagnose, and recover from errors*, and (10) *Help and documentation*.

As a summary, we can say that the literature on this issue is heterogeneous, coming from different sources that are aimed at different audiences, and that synthesizing all this information is not necessarily trivial. Some usability aspects are well covered by many authors (like clarity, consistency, etc.) but many authors omit aspects that are interesting when evaluating API usability or cover them only superficially.

Our approach to this problem is to follow a systematic method based on a comprehensive model of usability and context-of-use. From this comprehensive model we derived a set of heuristics, some of them are derived from the API literature but others are inferred for those usability attributes that were mainly ignored in the API literature. The objective is to obtain, for each relevant usability attribute identified in our comprehensive model, one or more heuristics, which are then typically mapped to several specific guidelines. Therefore, this work is complementary with the metric approach of authors like Scheller and Kühn [46], facilitating the realization of heuristic and subjective studies.

2.2. *The proposed usability model*

The usability study in this paper is based on the usability taxonomy by Alonso-Ríos et al. [1], as, among the expanded models mentioned before, it is the most complete one that has been published in full. The stated goals for this taxonomy were ([3], p. 586):

- To be comprehensive, covering all the usability aspects from the literature but avoiding contradictions and redundancy.
- To be structured hierarchically into several levels of detail. Typically, the usability models in the literature only have one level.
- To be applicable to any type of product. This contrasts strongly with traditional usability models, which are restricted to IT systems.
- To provide definitions for all the attributes and subattributes.

We give a brief overview of the taxonomy below. In Fig. 1 we can see the first levels of the taxonomy. These levels are further expanded in more attributes and subattributes. The complete version is in Alonso-Ríos et al. [1].

The first-level attributes are:

- *Knowability*: the property by means of which the user can understand, learn, and remember how to use the system. This attribute is subdivided into *clarity, consistency, memorability*, and *helpfulness*. The first three apply to *formal* (e.g., visual, acoustic, etc.) and *conceptual* aspects, and to the *functioning* of *user* and *system tasks*.
- *Operability*: the capacity of the system to provide users with the necessary functionalities and to permit users with different needs to adapt and use the system. This attribute is subdivided into *completeness, precision, universality* (e.g., *accessibility* and *cultural universality*), and *flexibility* (e.g., *controllability* and *adaptiveness*).
- *Efficiency*: the capacity of the system to produce appropriate results in return for the resources that are invested. The taxonomy draws a distinction between efficiency *in human effort, in task execution time, in tied up resources*, and *in economic costs*, with each category further decomposed into more subattributes.
- *Robustness*: the capacity of the system to resist error and adverse situations. The taxonomy draws a distinction between robustness *to internal error, to improper use, to third party abuse*, and *to environment problems*.

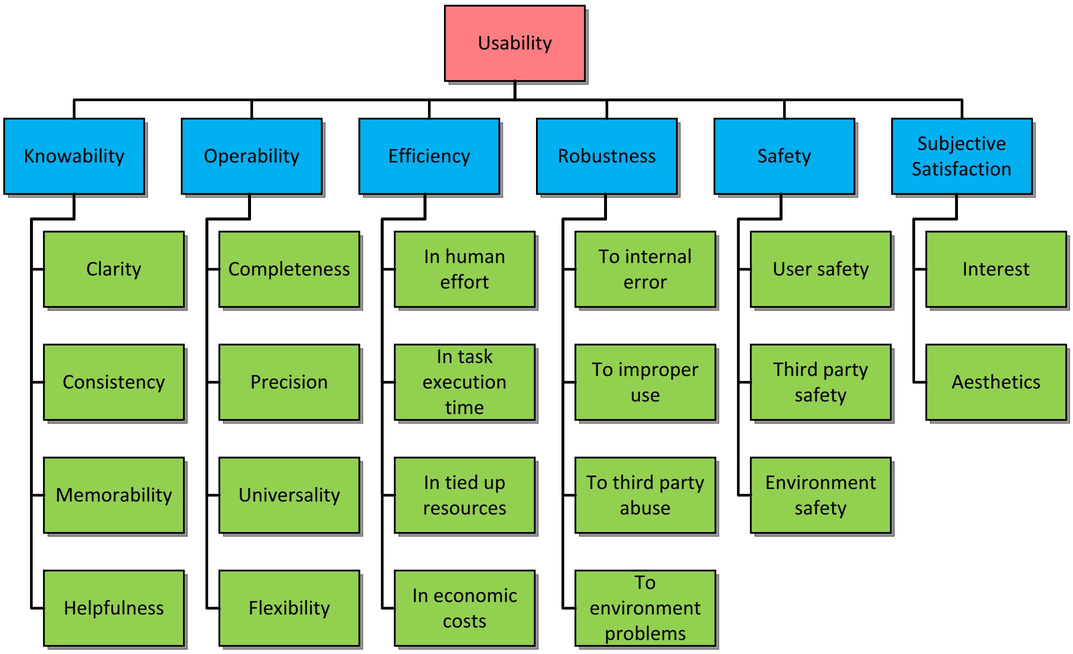


Fig. 1. First levels of the usability taxonomy.

- *Safety*: the capacity to avoid risk and damage derived from the use of the system. The taxonomy draws a distinction between *user safety, third party safety*, and *environment safety*. The first two are further subdivided into *physical safety, legal safeguarding, confidentiality*, and *safety of assets*.
- *Subjective satisfaction*: the capacity of the system to produce feelings of pleasure and interest in users. This attribute is subdivided into *interest* (the capacity to capture and maintain the attention) and *aesthetics* (the capacity of the system to please its user in sensorial terms).

As stated explicitly in the ISO 9241–11 definition of usability – and by researchers like Scheller and Kühn [46] – usability depends on the specific context of use. That is, the particular characteristics of the *users, tasks*, and *environments*. The expanded usability model [1] is complemented with a separate context-of-use taxonomy [2] that follows the same approach and principles as the usability taxonomy. That is, it was motivated by the lack of consensus on the meaning of the term and was intended as a detailed synthesis of the literature. The context of use is not something to be evaluated, but rather, is used to identify those categories that are more relevant to take into account in a usability study and that can also be used to interpret the usability results and decide which aspects to focus on.

The context-of-use taxonomy has the first-level attributes depicted in Fig. 2. At this broad level, the attributes are those commonly accepted in the literature on usability, namely, the system *users, tasks* performed by the users, and the *environments* in which the system is used. They are described in detail next.

- *User*: A user is a person who interacts directly or indirectly with the system. This attribute is subdivided into *role, experience, education, attitude to the system, physical characteristics,* and *cognitive characteristics*.
- *Task*: A task is a piece of work that the user carries out by interacting with the system. This attribute is subdivided into *choice in system use, complexity, temporal characteristics, demands, workflow controllability, safety, and criticality*.
- *Environment*: The environment consists of the external factors that affect the use of the system. It is distinguished between the *physical environment* (the surroundings and space in which the user operates the system), the *social environment* (the people with whom the user interacts and who effect the user's interaction with the system), and the *technical environment* (the technical equipment and infrastructures that support the functioning of the system).

2.3. *Field of application*

The field of application of our case study is sleep medicine, for which our research group has developed a decision support system for aiding clinicians to make decisions on disorders such as sleep apnea.

The basic physiological test needed to diagnose sleep diseases is the polysomnography. A polysomnography can be described as the comprehensive recording of biophysiological signals during sleep. These signals typically include electroencephalography (EEG), electrooculography (EOG), and electromyography (EMG).

These kinds of signals can be used to identify the distinct stages of a patient's sleep. The graphical and chronological representation of the sleep stages of a given patient is called a “hypnogram” and is constructed following the rules of the American Academy of Sleep Medicine (AASM[2]) [10].

Additional signals can be included in a polysomnography to detect some afflictions. For example, respiratory airflow, respiratory effort, and peripheral pulse oximetry, can be added in order to determine the existence of a Sleep Apnea-Hypopnea Syndrome (SAHS). This syndrome consists in the periodic, involuntary occurrence of pauses of airflow in the respiratory tracts for at least ten seconds.

The tool chosen for our case study is a medical decision-support system (DSS) for the diagnosis of SAHS [4]. It can be considered a comprehensive tool in that it classifies the sleep stages, analyzes respiratory activity, and provides detailed explanations of its results. This comprehensive approach contrasts with previously-existing tools that

[2] The American Academy of Sleep Medicine (AASM) is a sleep medicine association for professionals dedicated to the treatment of sleep disorders.

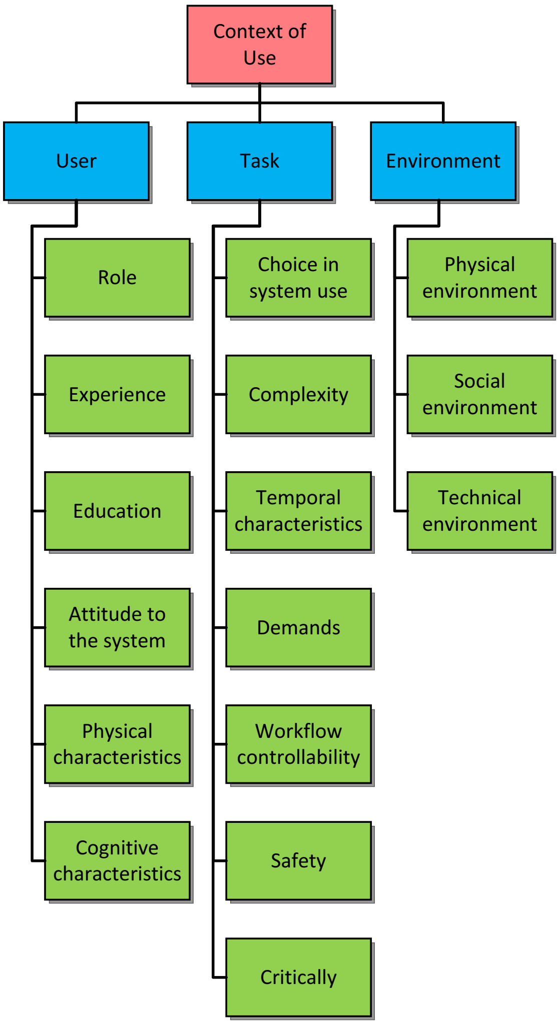


Fig. 2. First levels of the context-of-use taxonomy.

are typically focused on specific subtasks like patient screening, analysis of respiratory activity, and classification of apneic event types (Álvarez-Estévez & Moret-Bonillo, [6]).

Our objective in this paper is to assess the API of the DSS for the diagnosis of SAHS using different usability techniques. Our idea was not only to evaluate the API of our tool but also to obtain an extensive and hierarchically-organized set of heuristics and guidelines that can be easily generalized to be used in APIs of different tools and domains.

The idea of focusing on the API instead on the graphical interface is due to the particular domain in which this application works. SAHS is one of the most important sleep disorders, others are insomnia and restless legs syndrome or RLS (for a complete standard classification of sleep disorders the user is referred to the American Academy of Sleep Medicine [7]). For that reason there are many research prototypes and commercial tools available that address the SAHS problem (Álvarez-Estévez & Moret-Bonillo, [6]).

The strength of our tool resides in its analysis capabilities and one promising approach is to try to integrate its artificial intelligence features into an existing commercial or research development. But in order to do that we need to expose an interface for programming (i.e., the API) and, since the source code was originally created without plans to make it available to third parties, it is now necessary to carry out an analysis to ensure that this programming interface is usable.

3. Methodology

The first task we have taken was to conduct an extensive review of the literature on the usability of APIs (already explained in the background section). The literature is mainly focused on suggesting guidelines, best practices, and so on.

As we have seen, some authors connect their API usability criteria to the wider usability literature, whereas other authors do not. For example, the cognitive dimensions of Clarke [16] are based on the cognitive dimensions framework proposed by Green and Petre [25]; or the guidelines proposed by Myers and Stylos [40] that follow Nielsen's heuristics [43].

We argue that it is important to connect the API literature to a usability model, in order to ensure that they are addressing the same thing. In order to achieve this, we used the previously discussed usability taxonomy by Alonso-Ríos et al. [1], searching for correspondences between its usability attributes and the information in the literature on API usability.

Establishing correspondences between different usability classifications also allows us to validate them against each other. This is similar to how Winter et al. [52] investigated the validity of their usability model by comparing it against the ISO 15005:2002 standard ([31]). Their goals were, firstly, to validate their usability model by showing that it can be used to model the principles contained in the standard, and, secondly, to use their model to uncover deficiencies in the ISO standard. As a result, they identified problems like incompleteness, lack of explicitness, and inconsistency in the requirements and recommendations proposed by the standard.

We structured the guidelines and recommendations in the literature along two dimensions. Firstly, we mapped them to the corresponding usability attribute or subattribute in the aforementioned taxonomy. Secondly, we classified them into two categories, namely, heuristics and specific guidelines. Following Nielsen's terminology [43], a heuristic is defined as a broad rule of thumb. Guidelines, on the other hand, "can range from highly specific prescriptions to broad principles" ([32], p. 487).

In our case, heuristics are rules of thumb that can be applied to diverse situations, whereas guidelines are more specific examples of application of these heuristics in specific environments (always trying not to be very specific, maintaining a certain degree of generality).

Therefore, we took every item of information from the literature and, regardless of how it was initially formulated, we expressed it or rewrote it as a heuristic or a guideline. We also identified relevant usability attributes that were not being addressed by the literature and proposed requirements and heuristics for them.

For example, in the literature we can find guidelines like "avoiding cryptic abbreviations in names", "names should not obscure intent", "short variable names should not be used for big scopes", etc. All these guidelines can be summarized into a more general heuristic that states: "Names should be self-explanatory". These heuristics and guidelines are mapped to the usability subattribute of "Knowability – Clarity – Clarity of Elements".

The ultimate goal is to integrate all this information into the typical methods of a usability study – specifically, heuristic evaluation and guideline review. Examples of other usability study methods applied to APIs can be found in Myers et al. [41]. The API usability heuristics and guidelines identified are presented in detail in the next section.

Our case study consisted in a usability study of the API of a sleep medicine decision support system as it is going to be used by third parties. In this study we identify the following roles:

- *Usability engineers*, Computer science engineers and PhDs specialized in usability and with previous experience in health informatics.
- *API developers*, Computer science engineers in charge of maintaining

the sleep medicine API.

- *API users*, Computer science engineers specialized in health and who will use the API in real systems.

As mentioned in the literature review on API usability, the existing heuristics and guidelines in the literature have been typically integrated into well-known usability activities such as heuristic evaluation or interviews. These techniques are complementary and offer different kinds of insights – for example, heuristic evaluation is performed by usability experts that check conformance with good practices, whereas interviews involve the subjective judgments of the users. We chose the following activities for our study:

- *Requirements analysis*, prior to the design, to identify the functionalities that the software should have.
- *Context-of-use analysis*, prior to the design to analyze the influence of the context-of-use in our usability analysis.
- *Heuristic evaluation*, which offers "rules of thumb" to guide the design combined with guideline review, to check that the implementation conforms to precise guidelines.
- *Subjective analysis*, composed of user questionnaires (to assess the subjective opinions of the users), and interviews (to elicit more detailed information from the users on specific aspects).

The specific characteristics and the results of these usability analyses are discussed in detail in Section 5 (Case Study).

4. Designing API usability heuristics and guidelines

4.1. Mapping between usability taxonomy and the API literature

The first step in developing API usability heuristics and guidelines is to compare the usability attributes of our methodology (those derived from the usability taxonomy) with the usability attributes proposed by other authors.

In this respect, we compared our usability attributes with those of Clarke [16] based on cognitive dimensions, as these cognitive dimensions attempt to cover all the usability related aspects. In the same way we compared our work with Nielsen heuristic attributes [43] that were used by Myers and Stylos to develop API usability guidelines.

Finally, we also choose the work of Zibran [53] that described a detailed set of 22 specific guidelines after an exhaustive study of existing literature. His work is not intended to be methodological, for example, he described five characteristics that an API should present to be usable but he does not relate these characteristics to the proposed guidelines. Nevertheless, Zibran's work was very comprehensive so it will be interesting to compare his guidelines with the usability attributes derived from our taxonomy.

Therefore Table 1 shows the correspondences between the usability taxonomy by Alonso-Ríos et al. and those key authors from the literature on API usability. Each second-level of the usability taxonomy is mapped to one or more usability criteria (or recommendations, guidelines, etc.) for APIs proposed by other authors. It should be considered that in this mapping we are dealing with natural language sentences, thus there is some degree of subjectivity in it.

Other authors in the literature, such as Jacques [33], Bloch [14], and Martin [37] do not propose usability attributes, so they were used to complete the heuristics and guidelines proposed in the next subsection.

The first thing that becomes clear is that all the criteria (or guidelines, etc.) for APIs that were proposed by other authors could be successfully mapped to an attribute (or, sometimes, several attributes) from the usability taxonomy. This has important consequences for both ends of the mappings. Firstly, it shows that the taxonomy is comprehensive enough to encompass all the information in the API literature even though the former is general-purpose and the latter is extremely specialized. Secondly, all the criteria in the API literature can be traced back to generic usability attributes derived from the usability literature. A failure to achieve any of these two things would mean that the usability taxonomy is incomplete or that the API literature is addressing something that is unrelated to usability.

However, not all authors cover the same usability attributes, or in the same way. Most noticeably, the criteria by Zibran are more comprehensive than the ones by Clarke, but Clarke also covers things that the others do not. This is why the synthesis pursued in this research is important – the literature on API usability is not only heterogeneous in presentation, but also in content.

It is also interesting to see that several usability attributes and subattributes in Table 1 (first column) do not have a correspondence with the usability attributes or guidelines proposed by the other authors included in the table (columns 2–5). We can see this in some *efficiency* subattributes (*efficiency in tied-up resources* and *efficiency in economic costs*), in some robustness subattributes (*robustness to third-party abuse* and *robustness to environment problems*) and in all the subattributes related to *safety* and *subjective satisfaction.*

However this is not necessarily a symptom of deficiencies in the literature and it is necessary to look at this on a case by case basis. Because the taxonomy is general-purpose, it will typically include attributes that are not relevant for a given domain. For example, for the five senses, only the visual aspects are applicable to APIs. But we have to take into account that this is only a first step in our analysis and will be examined in detail in next subsection (Section 4.2).

4.2. Proposed API usability heuristics and guidelines

The next step is to analyze all the sources of information that discuss API usability and to try to unify heuristics and guidelines for those usability attributes covered by many authors, to add depth if necessary, to resolve contradictions, and to try to develop new heuristics and guidelines for those usability attributes that were mainly ignored in the literature.

Tables 2–7 present the usability heuristics and guidelines obtained from our synthesis of the literature. As mentioned, they are organized according to the usability taxonomy by Alonso-Ríos et al. [1]. For each relevant usability attribute in this taxonomy, one or more heuristics are included, which are then typically mapped to several specific guidelines. A particular heuristic or guideline can be taken directly from the literature, or synthesized from the work of different authors, or it can be a new one obtained from the usability taxonomy. Each heuristic indicates the sources from which it was derived, if any. It is interesting to note that, as we said in the previous subsection, usability attributes such as *Efficiency, Safety*, or *Subjective Satisfaction*, do not have a corresponding heuristic or guideline extracted from the literature, so we have to create new ones just to complete the analysis based on the taxonomy.

We have also addressed some contradictions in the literature. For example:

- Some guidelines stated that a code that is no longer used should be deleted, that is, dead code [37], but other guidelines state that backwards compatibility is important [53].
- The contrast between the flexibility of having different ways of doing a task compared with the complexity of having too many options [53].
- A user study found that the Factory Pattern imposes difficulties on the programmers that naturally expect to use constructors to instantiate objects ([21]), but another author [51] states that you facilitate an API's future evolution when you expose a factory method rather than a constructor.
- Stylos and Clarke [49] stated that a create-set-call pattern is easier to use than constructors with required parameters but Piccioni et al. [44] did not find evidence for this. Scheller and Kühn [46] stated

Table 1
Usability taxonomy correspondences.

Attribute	Clarke [16]	Zibran [53]	Scheller & Khun [46]	Myers & Stylos [40] Nielsen [43]
Knowability				
Clarity	11. Role expressiveness 7. Penetrability	2. Naming 3. Ignorance of caller's perspective 10. Data types 11. Use of attributes 16. Long chain of reference 22 Intelligibility of source code	4. Classes for different tasks. 6. The naming of API elements. 8. Methods are very hard to find. 9. Method parameters that are self-explaining. 11. Strings should not be used if a better type exists. 12. Common usage scenarios. 22. API that exposes functionality through annotations.	1. Visibility of system status (Feedback) 2. Match between system and the real world. 5. Error prevention 6. Recognition rather than recall 8. Aesthetic and minimalist design
Consistency	10. Consistency	2. Naming 5. Consistency and conventions 7. Method parameters and return type	14. Parameters should be ordered consistently across methods.	2. Match between system and the real world. 4. Consistency and standards
Memorability	2. Learning style 12. Domain correspondence	1. Complexity 7. Method parameters and return type	5. The number of method parameters and return values has a strong influence on usability. 13. Functions with multiple consecutive parameters.	2. Match between system and the real world.
Helpfulness		4. Documentation and code examples 20. API Evolution or Change 22 Intelligibility of source code		9. Help users recognize, diagnose, and recover from errors 10. Help and documentation
Operability				
Completeness	8. API elaboration			
Precision		6. Conceptual correctness 10. Data types	19. Ambiguous method overloads. 20. Different overloadings of the same method	
Universality				
Flexibility	5. Progressive evaluation 6. Premature commitment 9. API Viscosity	8. Parameterized constructor 17. Implementation vs. interface dependency 19. Technical mismatch 20. API Evolution or Change 21. API Aging	1. An API should be minimal. 18. Fields should not be public. 23. Create-set-call pattern vs. constructors.	3. User control and freedom
Efficiency				
Human effort	1. Abstraction level 3. Working framework 4. Work-step unit	1. Complexity 15. Multiple ways to do one thing	2. Not be required to do anything the module could do itself. 3. Explicitly instantiate more than one type. 7. Code completion. 16. The abstraction level of an API. 17. The factory pattern. 21. Providing different API variants.	
Task execution time		9. Factory pattern 14. Leftovers for client code		7. Flexibility and efficiency of use
Tied-up resources				
Economic costs				
Robustness				
Internal error				
Improper use		7. Method parameters and return type 12. Concurrency 13. Error handling and exceptions 18. Memory management	10. The API should report usage errors. 15. Exceptions should only be used to indicate exceptional conditions.	5. Error prevention 9. Help users recognize, diagnose, and recover from errors
Third-party abuse				

(*continued on next page*)

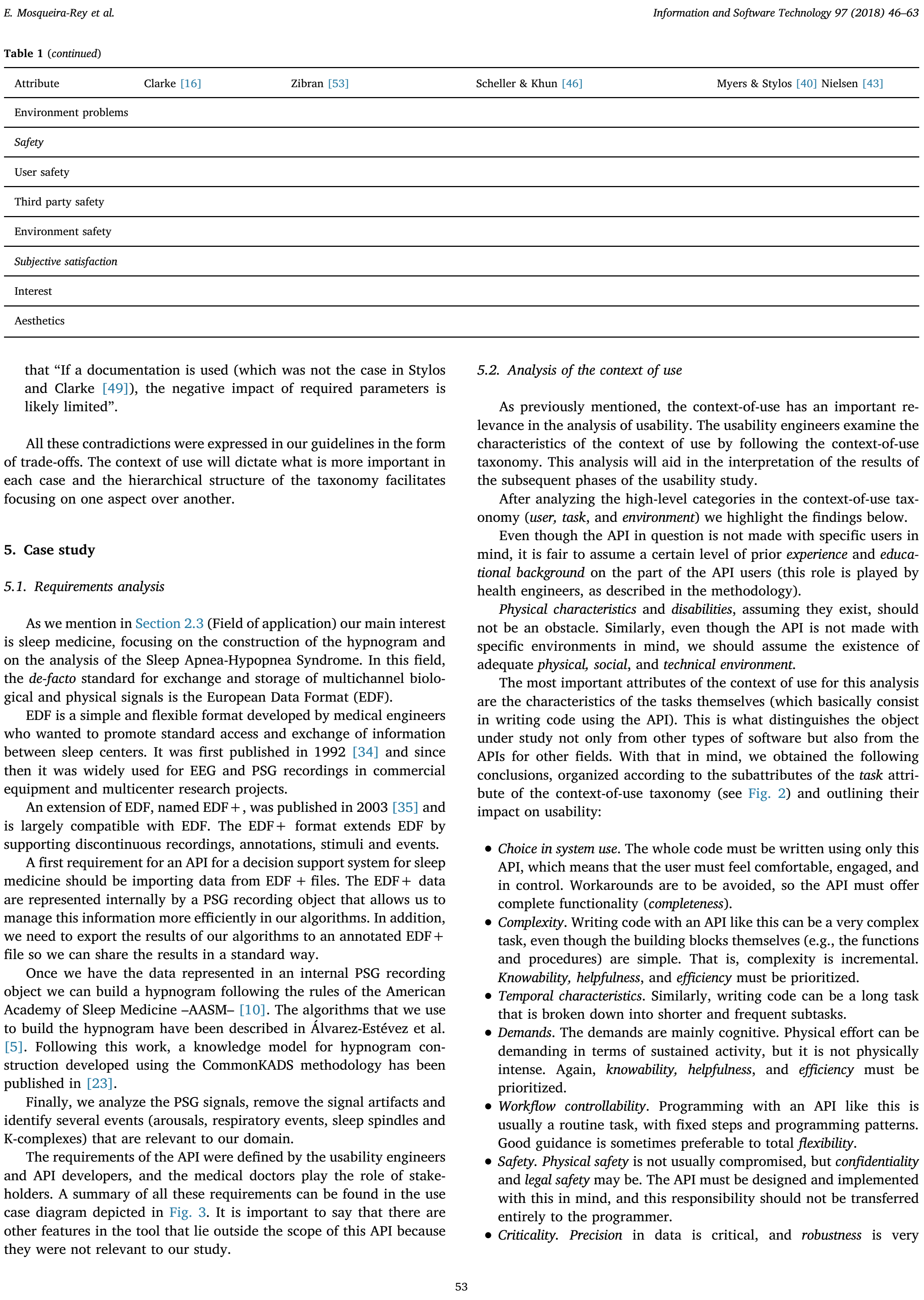


Table 1 (*continued*)

Attribute	Clarke [16]	Zibran [53]	Scheller & Khun [46]	Myers & Stylos [40]	Nielsen [43]
Environment problems					
Safety					
User safety					
Third party safety					
Environment safety					
Subjective satisfaction					
Interest					
Aesthetics					

that "If a documentation is used (which was not the case in Stylos and Clarke [49]), the negative impact of required parameters is likely limited".

All these contradictions were expressed in our guidelines in the form of trade-offs. The context of use will dictate what is more important in each case and the hierarchical structure of the taxonomy facilitates focusing on one aspect over another.

5. Case study

5.1. Requirements analysis

As we mention in Section 2.3 (Field of application) our main interest is sleep medicine, focusing on the construction of the hypnogram and on the analysis of the Sleep Apnea-Hypopnea Syndrome. In this field, the *de-facto* standard for exchange and storage of multichannel biological and physical signals is the European Data Format (EDF).

EDF is a simple and flexible format developed by medical engineers who wanted to promote standard access and exchange of information between sleep centers. It was first published in 1992 [34] and since then it was widely used for EEG and PSG recordings in commercial equipment and multicenter research projects.

An extension of EDF, named EDF+, was published in 2003 [35] and is largely compatible with EDF. The EDF+ format extends EDF by supporting discontinuous recordings, annotations, stimuli and events.

A first requirement for an API for a decision support system for sleep medicine should be importing data from EDF + files. The EDF+ data are represented internally by a PSG recording object that allows us to manage this information more efficiently in our algorithms. In addition, we need to export the results of our algorithms to an annotated EDF+ file so we can share the results in a standard way.

Once we have the data represented in an internal PSG recording object we can build a hypnogram following the rules of the American Academy of Sleep Medicine –AASM– [10]. The algorithms that we use to build the hypnogram have been described in Álvarez-Estévez et al. [5]. Following this work, a knowledge model for hypnogram construction developed using the CommonKADS methodology has been published in [23].

Finally, we analyze the PSG signals, remove the signal artifacts and identify several events (arousals, respiratory events, sleep spindles and K-complexes) that are relevant to our domain.

The requirements of the API were defined by the usability engineers and API developers, and the medical doctors play the role of stakeholders. A summary of all these requirements can be found in the use case diagram depicted in Fig. 3. It is important to say that there are other features in the tool that lie outside the scope of this API because they were not relevant to our study.

5.2. Analysis of the context of use

As previously mentioned, the context-of-use has an important relevance in the analysis of usability. The usability engineers examine the characteristics of the context of use by following the context-of-use taxonomy. This analysis will aid in the interpretation of the results of the subsequent phases of the usability study.

After analyzing the high-level categories in the context-of-use taxonomy (*user, task*, and *environment*) we highlight the findings below.

Even though the API in question is not made with specific users in mind, it is fair to assume a certain level of prior *experience* and *educational background* on the part of the API users (this role is played by health engineers, as described in the methodology).

Physical characteristics and *disabilities*, assuming they exist, should not be an obstacle. Similarly, even though the API is not made with specific environments in mind, we should assume the existence of adequate *physical, social*, and *technical environment.*

The most important attributes of the context of use for this analysis are the characteristics of the tasks themselves (which basically consist in writing code using the API). This is what distinguishes the object under study not only from other types of software but also from the APIs for other fields. With that in mind, we obtained the following conclusions, organized according to the subattributes of the *task* attribute of the context-of-use taxonomy (see Fig. 2) and outlining their impact on usability:

- *Choice in system use*. The whole code must be written using only this API, which means that the user must feel comfortable, engaged, and in control. Workarounds are to be avoided, so the API must offer complete functionality (*completeness*).
- *Complexity*. Writing code with an API like this can be a very complex task, even though the building blocks themselves (e.g., the functions and procedures) are simple. That is, complexity is incremental. *Knowability, helpfulness*, and *efficiency* must be prioritized.
- *Temporal characteristics*. Similarly, writing code can be a long task that is broken down into shorter and frequent subtasks.
- *Demands*. The demands are mainly cognitive. Physical effort can be demanding in terms of sustained activity, but it is not physically intense. Again, *knowability, helpfulness*, and *efficiency* must be prioritized.
- *Workflow controllability*. Programming with an API like this is usually a routine task, with fixed steps and programming patterns. Good guidance is sometimes preferable to total *flexibility*.
- *Safety. Physical safety* is not usually compromised, but *confidentiality* and *legal safety* may be. The API must be designed and implemented with this in mind, and this responsibility should not be transferred entirely to the programmer.
- *Criticality. Precision* in data is critical, and *robustness* is very

Table 2
Knowability heuristics.

Id.	Heuristics	Guidelines	Sources
Knowability – Clarity – Clarity of Elements			
KCE-1	Names should be self-explanatory.	• Cryptic abbreviations and names should be avoided. • Be as expressive as possible, do not obscure intent (e.g. using Hungarian notation, etc.). • Names should be self-documenting. • If you have to look at the implementation (or documentation) of the function to know what it does, then you should work to find a better name or rearrange the functionality. • Short variable names can be used for tiny scopes (i, j), but for big scopes you should use longer names.	[13][16][26] [33][37][40] [45][46][53]
KCE-2	Data types should be as specific as possible to make the code more readable.	• Avoid generic types when a specific type can be used. • Avoid boolean or string types. Use an enumeration value instead: list.insert("value", true) vs. list.insert("value", Insert.ORDERED) • Another option is avoiding the parameter by splitting a large function into several smaller functions: insert(“value”), insertOrdered(“value”)	[13] [16] [26] [33] [37] [46] [53]
Knowability – Clarity – Clarity of Structure			
KCS-1	Inheritance hierarchies should not be too deep.	• The inheritance tree should not be unnecessarily deep. • However, the number of children (classes that inherit directly from a particular class) does not have a significant effect on the understandability of the hierarchy.	[26] [24] [53]
KCS-2	When reading code that uses the API, it should be easy to understand what that code does.	• Make the code easy to read. • The code should flow in a clear way. • When method calls (or other elements) are chained together, the resulting code can be quickly understood. But avoid long chains of method delegations that are difficult to track. • Encapsulate recurrent and complex conditional sentences in functions: if (shouldBeDeleted(timer)). • Avoid negative conditionals. • There should be a correspondence between naming and structure (e.g. use “get” for reading values and “set” to writing).	[16] [33] [37] [40] [53]
KCS-3	Do not expose core API functionality through secondary elements (attributes, annotations, etc.).	• Many developers will not expect the core functionality of an API to be controlled through secondary elements like attributes or annotations. • Combinatorial effects between different elements should be avoided. • If they cannot be avoided, the relationship between different elements should be expressed through good naming.	[16] [46] [53]
KCS-4	If the API is open source, the internal implementation should be also readable.	• Use proper indentation following a standard consistently. • Do not use long source files, classes, methods, etc. • Make the code readable (e.g. loop conditions). • Put classes and methods that are most frequently used first. • Group together methods representing related tasks.	[33] [53]
KCS-5	The API should be loosely coupled.	• Elements that don't depend upon each other should not be artificially coupled. • If some elements are coupled together, this coupling should be obvious, e.g. temporal couplings between functions should be clear through arguments: IntermediateResult ir = doFirst(); FinalResult fr = doLast(ir);	[37]
KCS-6	The different API elements (classes, methods, etc.) should be placed in the most logical place to be and where users expect to find them.	• Place classes in packages in a coherent manner: Group classes with similar functionalities or that are used together in the same package. Classes for different tasks should be placed in different packages. • In an object oriented environment methods and functions should be bundled with the data needed for their operation.	[45] [46] [50]
Knowability – Clarity – Clarity in Functioning			
KCF-1	Functions should focus on doing one thing.	• Functions should do one thing. • Do not include output arguments as they are counterintuitive. Use the return argument or change the state of the owning object.	[37]
KCF-2	Functions should perform only the tasks described in their names.	• Principle of Least Astonishment: User of API should not be surprised by behavior. • The methods should not have side-effects. • If side-effects are present, they should be described in the function's name.	[13] [26] [37]
KCF-3	When writing code it should be easy to know what classes and methods of the API to use.	• The methods and classes in the API are well defined. • The names of classes and methods, their structure and organization ease the understanding of their functioning. • Code should be placed where a reader would naturally expect it to be.	[16] [37], [40]

(*continued on next page*)

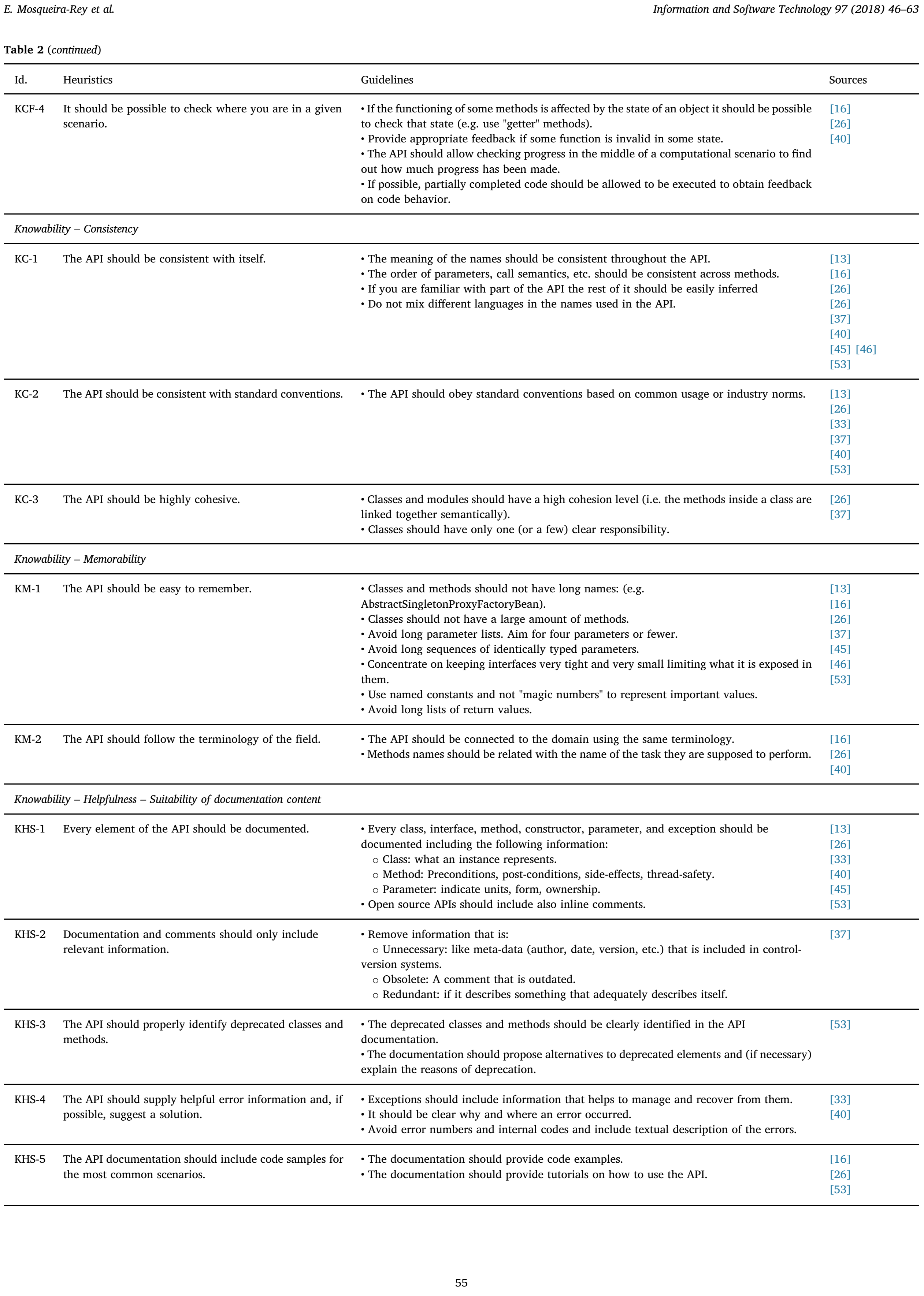


Table 2 (*continued*)

Id.	Heuristics	Guidelines	Sources
KCF-4	It should be possible to check where you are in a given scenario.	• If the functioning of some methods is affected by the state of an object it should be possible to check that state (e.g. use "getter" methods). • Provide appropriate feedback if some function is invalid in some state. • The API should allow checking progress in the middle of a computational scenario to find out how much progress has been made. • If possible, partially completed code should be allowed to be executed to obtain feedback on code behavior.	[16] [26] [40]
Knowability – Consistency			
KC-1	The API should be consistent with itself.	• The meaning of the names should be consistent throughout the API. • The order of parameters, call semantics, etc. should be consistent across methods. • If you are familiar with part of the API the rest of it should be easily inferred • Do not mix different languages in the names used in the API.	[13] [16] [26] [26] [37] [40] [45] [46] [53]
KC-2	The API should be consistent with standard conventions.	• The API should obey standard conventions based on common usage or industry norms.	[13] [26] [33] [37] [40] [53]
KC-3	The API should be highly cohesive.	• Classes and modules should have a high cohesion level (i.e. the methods inside a class are linked together semantically). • Classes should have only one (or a few) clear responsibility.	[26] [37]
Knowability – Memorability			
KM-1	The API should be easy to remember.	• Classes and methods should not have long names: (e.g. AbstractSingletonProxyFactoryBean). • Classes should not have a large amount of methods. • Avoid long parameter lists. Aim for four parameters or fewer. • Avoid long sequences of identically typed parameters. • Concentrate on keeping interfaces very tight and very small limiting what it is exposed in them. • Use named constants and not "magic numbers" to represent important values. • Avoid long lists of return values.	[13] [16] [26] [37] [45] [46] [53]
KM-2	The API should follow the terminology of the field.	• The API should be connected to the domain using the same terminology. • Methods names should be related with the name of the task they are supposed to perform.	[16] [26] [40]
Knowability – Helpfulness – Suitability of documentation content			
KHS-1	Every element of the API should be documented.	• Every class, interface, method, constructor, parameter, and exception should be documented including the following information: ○ Class: what an instance represents. ○ Method: Preconditions, post-conditions, side-effects, thread-safety. ○ Parameter: indicate units, form, ownership. • Open source APIs should include also inline comments.	[13] [26] [33] [40] [45] [53]
KHS-2	Documentation and comments should only include relevant information.	• Remove information that is: ○ Unnecessary: like meta-data (author, date, version, etc.) that is included in control-version systems. ○ Obsolete: A comment that is outdated. ○ Redundant: if it describes something that adequately describes itself.	[37]
KHS-3	The API should properly identify deprecated classes and methods.	• The deprecated classes and methods should be clearly identified in the API documentation. • The documentation should propose alternatives to deprecated elements and (if necessary) explain the reasons of deprecation.	[53]
KHS-4	The API should supply helpful error information and, if possible, suggest a solution.	• Exceptions should include information that helps to manage and recover from them. • It should be clear why and where an error occurred. • Avoid error numbers and internal codes and include textual description of the errors.	[33] [40]
KHS-5	The API documentation should include code samples for the most common scenarios.	• The documentation should provide code examples. • The documentation should provide tutorials on how to use the API.	[16] [26] [53]

Table 3
Operability heuristics.

Id.	Heuristics	Guidelines	Sources
Operability – Completeness			
OC-1	The API should provide the functionalities necessary to implement the tasks intended by the user.	• The API should satisfy user requirements. • All the possible scenarios should be taken into account. • The API should not require users to implement methods or create classes when it is not necessary.	[13] [16]
OC-2	The API should maintain backwards compatibility deprecating functions in a clear way.	• API changes should keep backwards compatibility. • Older functions can be highlighted as deprecated but should not be deleted unless it is clearly necessary. • However, private code that is no longer needed should be discarded (keeping dead code around is wasteful).	[33] [37] [53],
Operability – Precision			
OP-1	Numeric data types should be as precise as necessary.	• Use more precise data types: (e.g. double −64 bits- rather than float −32 bits-). • Do not use floating point if precision is needed (e.g. monetary values). Use specific types (e.g. Currency or Money) or use an integer type and round appropriately.	[13] [37], [53]
OP-2	Data types should be as conceptually precise as necessary.	• Do not use lists when a set is a better option (because lists allow duplicates and sets do not). • Use subtyping only when a *is-a* relationship is present (use composition instead). • In some occasions, it is better not to be too precise. Declaring a variable to be an ArrayList when a List will do is overly constraining. • When overloading, data types in parameters and return values should be precise to avoid ambiguities. If it is not possible to avoid ambiguities then it is better to use different method names.	[26] [45] [46] [53]
Operability – Universality			
OU-1	The API should avoid the use of elements (units, formats, spellings, etc.) that are not universally recognized.	• Avoid the use of expressions that are circumscribed to a particular language (use more universal expressions instead). • Avoid the use of units that are not standard or are outside the field of the API. • When there are no standards allow the use of the different possibilities (e.g. the first day of the week can be Monday or Sunday).	
Operability – Flexibility			
OF-1	The API should be easy to change.	• The implementation details should not "leak" into the API: e.g. Public classes should have no public fields. • Interfaces and abstract classes should be preferred over concrete classes (e.g. using List instead of ArrayList or LinkedList). • It should be easy to extend the API to perform new features: e.g. new classes. • Base classes should know nothing about their derivative classes. • Favor polymorphism: Preferring non-static methods to static methods. Using polymorphism instead of If/Else or Switch/Case.	[13] [16] [26] [33] [37] [46] [53]
Operability – Flexibility – Controllability – Workflow Controllability – Freedom in tasks			
OFC-1	The API should not force users to make irreversible decisions without all the information available.	• The user should have all the information available when making a decision. • If not all the information is available the situation should be reversible, e.g. using abstraction and allowing to change an ArrayList for a LinkedList. • Another possibility is to offer several ways of performing a task (e.g. a constructor with required parameters vs. a parameterless constructor) but documenting clearly the implications of each way.	[16] [46]
Operability – Flexibility – Controllability – Workflow Controllability – Reversibility			
OFC-2	The API should allow reverting actions and returning to a previous state.	• It should be possible to revert the state of an object to a previous state if necessary. For example: If not all the information is available before taking an irreversible decision, the situation should be reversible (e.g. creating and object with a default constructor and then using setters to modify the state).	[16] [26] [40] [53]

important too. Again, the API must be designed and implemented with this in mind.

5.3. Heuristic evaluation

The heuristic evaluation involves usability specialists examining a product and assessing its usability according to a set of good practices or guidelines. This is one of the most popular usability techniques because it can quickly and inexpensively detect significant usability problems during the earliest stages of a project. Heuristic evaluation is typically informal and best performed by several persons individually, as one person normally discovers only a fraction of the actual usability problems [42].

Here it is important to say that this software was developed by several programmers in a long period of time within different projects with different objectives in mind and was never intended to be used by third parties. So, it is a system that meets its functional requirements but it is not easy to use or integrate into third-party applications.

Table 4
Efficiency heuristics.

Id.	Heuristics	Guidelines	Sources
Efficiency – Efficiency in human effort / task execution time			
EH-1	The level of abstraction of the API should be adequate for the users and the domain.	• The level of abstraction exposed by the API should match the expectations of targeted developers. • Do not mix higher level concepts with lower level concepts in the same abstraction. • Names should be at the appropriate level of abstraction. • There should be a balance between the flexibility of having different ways of doing a task and the complexity of having too many options. • Do not use a factory pattern in situations in which a constructor would be more adequate.	[16] [21] [26] [33] [37] [46] [53]
EH-2	The API should require the user to type as little as possible.	• Common tasks should be completed in a single step or with minimum coding. • The user should not be forced to put too much effort when doing a task. • Reduce the need for boilerplate code. • The user error handling code should be minimal.	[13] [16] [26] [33] [46] [53]
EH-3	If the API must be complex, establish layers of complexity for beginners and advanced users.	• Establish different features for "beginners" and "advanced" users. • Use the Facade pattern to define a higher-level interface that makes the system easier to use without hiding the lower-level functionality from the ones that need it. • The API should minimize the amount of information that the users need in order to use it.	[8] [16] [46] [53]
Efficiency – Efficiency in task execution time			
ET-1	The API should not force the user to take actions that would affect performance.	• Avoid public mutable types as they may often require needless defensive copying. • Favor composition over inheritance because inheritance would tie the class forever to its superclass. • Favor the use of interfaces above implementation types as the latter tie you to a specific implementation.	[13] [40] [46]
Efficiency – Efficiency in tied-up resources			
ER-1	The API should not excessively occupy limited resources.	• The API should not incur in resource leaks in finite system resources (i.e. memory, file handles) that become exhausted by repeated allocation without release. • The API should not make excessive use of shared resources like processor, Internet connection, disk, memory, etc.	
Efficiency – Efficiency in economic costs			
EC-1	The economic costs derived from using the API (if any) should be reasonable.	• License costs should be adequate to the services offered and should be in line with market prices. • Using the API should not involve excessive personnel or equipment costs.	

The heuristic evaluation was carried out by the usability engineers. The inputs for this activity were the original API, the requirements, and the API heuristics and guidelines. The results of this evaluation were a list of strong and weak points of the usability of the API and a set of proposals for improvement.

The process was as follows: First the heuristics and guidelines of Tables 2–7 (which follow the attributes of the usability taxonomy) were used to check if they were fulfilled. Then we classified each one according to the following categories:

- *Yes*. It is fulfilled.
- *Partially*. It is only partially fulfilled (i.e. in some parts of the code it is fulfilled but in other parts it is not, or some guidelines are fulfilled but others are not, etc.)
- *No*. It is not fulfilled.
- *Not Applicable (N/A)*. It is not possible to apply the heuristic for various reasons (it refers to aspects that are not supported by the current programming language being used, it is not possible to apply it in the current development phase, etc.).

Finally, the results of analyzing the 44 heuristics proposed are depicted in Fig. 4.

As we can see there is an important dichotomy, heuristics were met or were not met, there is practically no partial compliance, which implies that programmers were careful to address some aspects, but others were completely left out.

The problems that we found in the API were mostly from the *knowability* attribute, for example about *clarity* we found that: names were not self-explanatory (KCE-1), it was not easy to understand what the code did (KCS-2), functions performed several tasks that not always were described in their names (KCF-1, KCF-2) and it was not clear what classes and methods we need to use to perform some tasks (KCF-3).

Also the API was not *consistent* with itself in several aspects –order of parameters, language used, etc. (KC-1)– and it was also not consistent with the conventions of the language used –Matlab (KC-2)–. Due to this the API had a low *memorability* (KM-1) and was difficult to use because it was poorly documented (KHS-1), did not provide helpful information in the case of an error (KHS-4) and did not provide code samples for the most common scenarios (KHS-5).

We found also problems in other categories like *operability* –the API was not easy to change (OF-1)–; *efficiency* –the user needed to put too much effort when doing a task (EH-2), and did not use the resources efficiently (ER-1)–; *robustness*, –the API did not use exceptions to manage errors (RU-1) and exposed vulnerabilities that would allow users to make errors (RU-3)–; and finally *subjective satisfaction*, –Using the API was not satisfying (SI-1)–.

We can see that many of the categories that were identified as important by the context of use are not fulfilled by the code after the heuristic analysis. For example, the code is not easy to understand since it has some important drawbacks in terms of clarity, consistency and

Table 5
Robustness heuristics.

Id.	Heuristics	Guidelines	Sources
Robustness – Robustness to internal error			
RI-1	The API should not have bugs in its functioning.	• The functioning of the API should be correct and not include catastrophic bugs. For example: ○ Arithmetic bugs (i.e. Division by zero). ○ Logic bugs (i.e. infinite loops). ○ Resource bugs (i.e. null pointer dereference, using uninitialized variables). ○ Multi-threading bugs (i.e. deadlocks).	
Robustness – Robustness to improper use / third-party abuse			
RU-1	The API should allow detecting and managing errors without breaking the execution or leaving the error undetected.	• Exceptions should be handled near to where they occurred. • The API should generate an error as soon as possible after an improper use is detected. • The error messages should convey sufficient information. • Exceptions should be used when exceptional processing is demanded. • Prefer standard exceptions over own exceptions and specific exceptions instead more generic ones. • User input should be validated as it is entered. • Code assertions should be used to avoid an invalid state to occur. • Use mechanisms (if available) to detect errors at compile time and not at runtime (concrete data types, generics, strong typing, etc.).	[13] [26] [40] [33] [45] [46] [53]
RU-2	The API should facilitate managing non common but correct situations without generating exceptions or forcing users to catch them.	• The API should not demand unnecessary exceptional processing: For example, do not return null values, use instead a zero-length array, an empty collection or an "optional" value if supported by the language. • The API should not use exceptions for non-exceptional situations (e.g. reaching the end of a list).	[13] [46] [53]
RU-3	The API should not expose vulnerabilities that would allow users to make errors.	• Class members should be private unless there is good reason to expose them. • Objects that are building blocks / data types should be immutable objects. • The API should limit the state-space of mutable objects keeping it small and well-defined. • The API should take into account concurrent access if multiple threads are needed. • The API should not leave the tasks of memory management and garbage collection to the user because this is less usable and prone to error.	[13] [53]

memorability (all of them included into the *knowability* attribute).

The context has also helped us to detect situations that were, at first, classified as problematic but, after their review, they can be considered as lesser or minor problems. For example, efficiency is fundamental and the fact that the complete sleep analysis performed by the tool took a couple of minutes to complete would seem to be inefficient. But if we take into account the particular characteristics of that task, in which a polysomnography is analyzed off-line after a long night of sleep, then

Table 6
Safety heuristics.

Id.	Heuristics	Guidelines	Sources
Safety – User safety / Third-party safety – Legal			
SUL-1	The API should not put the user into legal trouble.	• The API should not grant access to data you have no permission to access (e.g. copyright protected content, confidential content, etc.). • If third-party elements are used, their license should explicitly state what you are allowed to do with them, e.g. CODECs. • Third-party safety (e.g. kids) should be assured.	
SUL-2	The API should clearly state its license of use.	• For example, stating whether it is open source, a GPL-type license, etc.	
Safety – User safety / Third-party safety – User confidentiality			
SUC-1	The API should not compromise the confidentiality of the users' personal information.	• It should be clear what personal data is being accessed (name, address, spatial geolocation, etc.) and obtain access only after asking for permission. • The API should not access other data stored in the user computer that is not necessary for its functioning. • The API should not give your personal data to third-parties without asking for permission. • There should be a balance between the security checks that are activated by default in the API and the flexibility of allowing deactivating these checks.	
Safety – User safety / Third-party safety – Safety of user assets			
SUA-1	The API should not compromise the security of the users' assets.	• For example: deleting files on your hard drive, making you lose money, compromising your intellectual property, etc.	

Table 7
Subjective satisfaction heuristics.

Id.	Heuristics	Guidelines	Sources
Subjective satisfaction – Interest/Aesthetics			
SI-1	Using the API should be satisfying.	• The API should be sufficiently engaging to avoid users losing interest in its use.	
		• The API should not be aesthetically displeasing, e.g. having weird names, using special characters in an inappropriate way, etc.	

these extra minutes are not very significant.

Therefore, after the heuristic analysis and the interpretation of its results with the data of the context of use, we decided to modify the API focusing mainly on the *knowability* problems that were easy to solve at this stage of development and that would be much more costly to solve later. Nevertheless problems in other categories were also addressed.

5.4. Subjective analysis

The existing API was modified to address the problems found during the heuristic evaluation. This modified API was used as input for the subjective analysis.

The usability engineers converted the heuristics and guidelines from Tables 2 to 7 into a questionnaire that could be easily filled out by the API users, and in which an attempt was made to simplify or clarify the more technical or confusing aspects so that they could be easily interpreted. This usability questionnaire completes the input to the subjective analysis.

Once the new API and the questionnaire had been developed the subjective analysis of the usability of the sleep medicine API was carried out in situ at the Medical Center of The Hague, The Netherlands (*Haaglanden Medisch Centrum* or HMC) where a prototype of the decision support system under analysis is currently in use.

The head of the computer service at the sleep center department played the role of API user and was the one in charge of carrying out

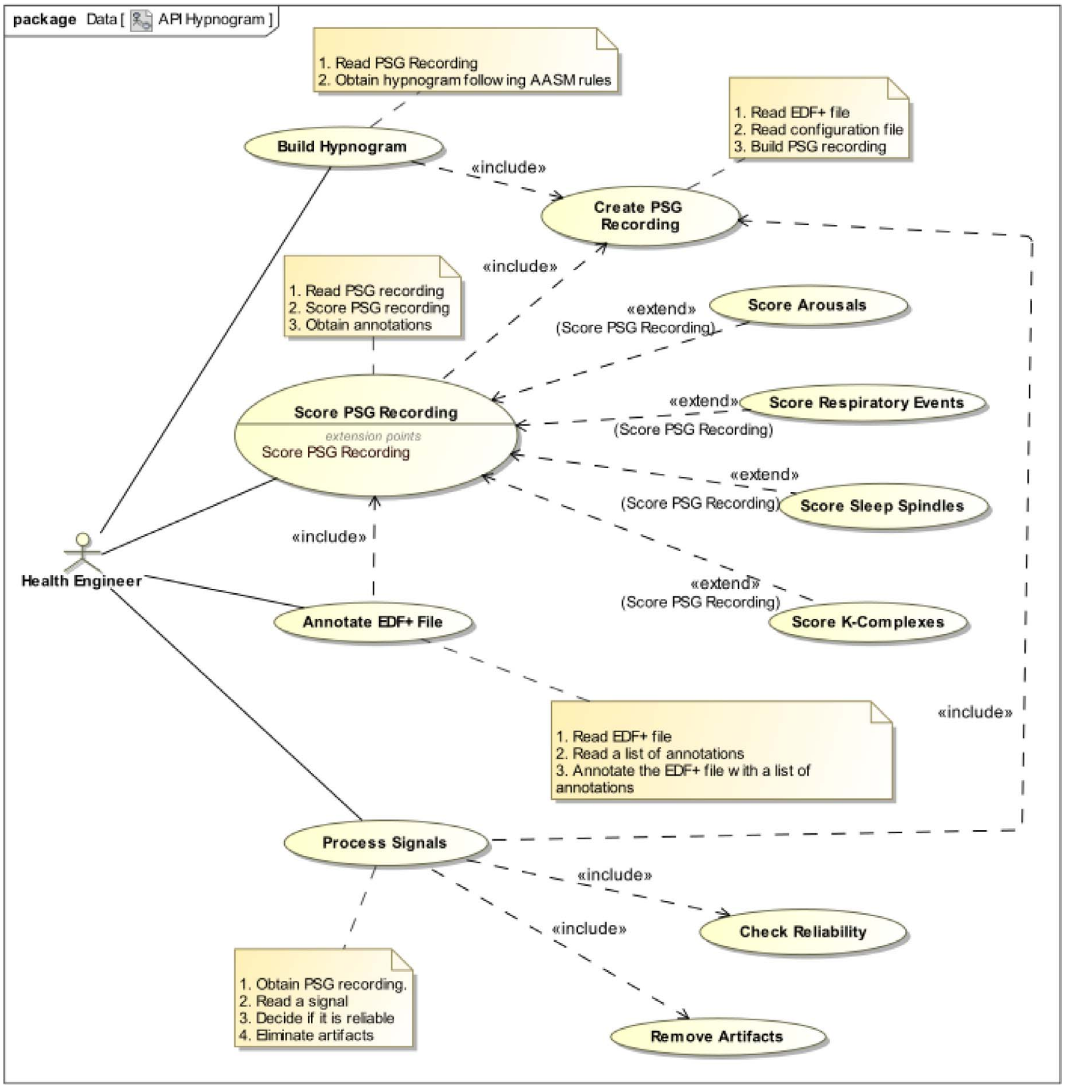


Fig. 3. Use case diagram identifying the main functionalities of the API under study.

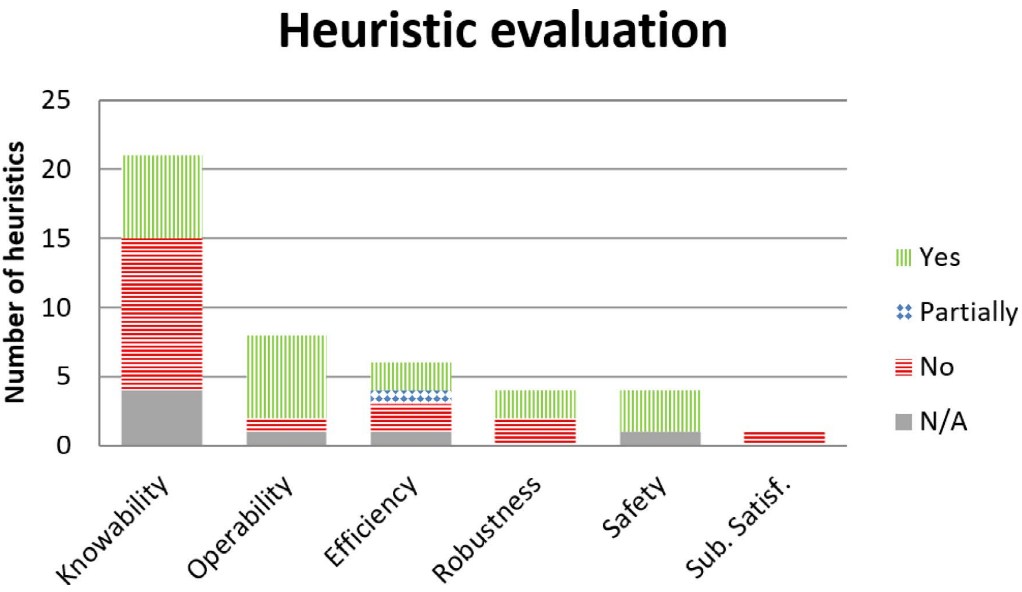


Fig. 4. Results of the heuristic evaluation (N/A: Not Applicable).

this analysis with the assistance of one of the API developers. It is important to emphasize that this subjective analysis was an initial evaluation to try to detect errors in the initial stages of the API design and that it is planned to make new subjective analyses with more users before setting the final design.

Once the API user had completed the questionnaire, the user had an interview with API developers to try to clarify any questions that could have had when filling the questionnaire. For example, the user answered several questions with a "*Not Enough Information"* answer. This is reasonable in part because some aspects like Efficiency or Robustness cannot be analyzed by the user easily without using extensively the API in real production environments. Other reason for having a *Not Enough Information* answer is that, in some questions, the questionnaire assumes that you have access to the source code of the API and this is something that is not always true. Nevertheless, a few answers of this type where due to possible ambiguities or misunderstandings that were clarified during the interview.

The answers of the questionnaire were analyzed by the usability engineers to obtain the final results of this analysis. A summary of them can be seen in Fig. 5.

As we can see, most of the heuristics within Knowability that were not fulfilled before the heuristic evaluation are now considered met by the user, if not completely at least partially. The only one considered unfulfilled was KHS-5: "*The API documentation should include code samples for the most common scenarios*". But it is something that is reasonable because in the current initial state of development implementing code samples was postponed until the API reaches a more stable state.

As we can see in Fig. 5, Efficiency aspects are still pending. For example, the user indicated as "No" in the questionnaire for heuristics EH-1: "*The level of abstraction of the API should be adequate for the users and the domain*" and EH-3: "*If the API is complex it should provide layers of complexity through progressive disclosure*". After the heuristic evaluation it was decided to raise the level of abstraction of the API to simplify the process for the users of the API and to eliminate some layers of complexity that previously existed (although only partially). But it turns out to be too high a level because the user was missing the access to some implementation details. This is an aspect in which it would be necessary to work more and also to gather the opinions of more users since it is a trade-off between simplicity and access to lower-level functionality.

Another question that was answered as a "No" by the user was SUL-2: "*The API should clearly state its license of use*". In the heuristic evaluation this questions was marked as a N/A because, since the software was an internal prototype, its license of use was not relevant for the development. Now, although the software is used under the shelter of a research project, since third parties are involved in the use and development of the software, the context of use has changed so now a clear software license should be included. As can be seen, context has a great importance when assessing usability. A situation that could be classified as a "minor problem" in a research prototype evolves to a "main concern" when that prototype evolves to a production stage.

Finally, we can see that there are still areas for improvement, for example, trying to identify and reduce heuristics that were classified as "Partially" or developing successive rounds of subjective analysis with new users, trying to involve them in the design of the API.

6. Discussion and conclusions

The heuristics and guidelines presented in this paper are the result of synthesizing a diverse set of publications from the literature on API usability. This process was guided and structured by a comprehensive usability taxonomy, and yielded the following findings about the literature:

- **The literature is formally heterogeneous**. Some authors present their criteria as attributes, whereas others simply offer recommendations (at different levels of specificity). Similarly, the literature comes from a great variety of sources (academic journals, magazines, websites, etc.) that are aimed at different audiences (researchers, developers, etc.). Synthesizing all this information is not necessarily trivial.
- **Content-wise, the literature is superficially more consistent**. After mapping the authors' criteria to the usability taxonomy, we conclude that all the authors tend to focus on the same main usability attributes, namely, *knowability, operability*, and *efficiency*. This consistency is only superficial, however. The criteria by Zibran and Grill are more comprehensive than the ones by Clarke, which omit important aspects related to, for example, *documentation* and data type *precision*. On the other hand, careful examination shows that Clarke also covers things that others do not. Zibran and Gill also

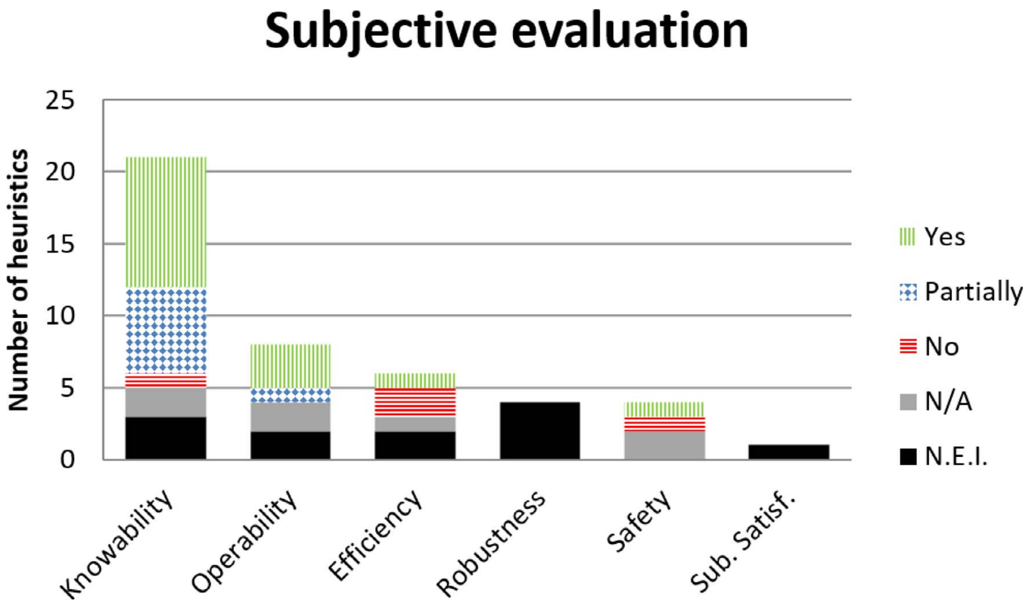


Fig. 5. Results of the subjective evaluation (N/A: Not Applicable. N.E.I.: Not Enough Information).

differ from other authors in that they attach importance to *robustness*. Classifying all this information according to the usability taxonomy simplified the task of integrating the work of these authors, identifying deficiencies at the finer levels of detail, and resolving contradictions.

- **The literature does not cover the full range of usability**. In particular, none of the authors has anything to say about *safety* or *subjective satisfaction*. The absence of the latter is especially noteworthy, as it is historically one of the main attributes of any usability model (see ISO 9241-11, [29]). It is true that, because usability is inherently subjective, most usability attributes also have a subjective component. But subjective satisfaction requires a category of its own because a user can like the aesthetics of a product but dislike everything else about it. We conclude that the literature on API usability is very technically-minded and tends to neglect the subjective component of usability. Our goal is to offer a more complete view of usability and to show how it is possible to synthesize the literature in a comprehensive list of heuristics and guidelines.
- **Metrics should be complemented with subjective aspects.** This technical vision of the usability is clear on those works that intend to analyze API usability through metrics. The idea of using metrics has some advantages, as you get objective values that can be obtained automatically and that allows you to measure the usability of your API without the need of experienced evaluators. But it has also several drawbacks, as you measure only aspects that are measurable, and you are leaving out subjective aspects that are inherent to every usability study. We think that our approach is not contradictory but complementary with metrics. Our taxonomy-derived criteria ease the always necessary usability assessment with human evaluators and users.
- **The literature neglects the context of use**. The context of use has a great significance in every usability analysis, but this aspect is only taken into account by a few authors. Our methodology includes a comprehensive analysis of the context of use that is used along with a comprehensive taxonomy of usability.

In light of these limitations, we proposed a new set of heuristics and guidelines. Some of these are a synthesis and refinement of the existing literature, whereas others address usability aspects that are neglected by the literature. These neglected aspects, classified according to the attributes of the usability taxonomy, would include:

- *Universality* (e.g., the API should avoid the use of elements that are not universally recognized).
- *Efficiency in tied-up resources* (e.g., the API should not excessively occupy resources that are limited).
- *Efficiency in economic costs* (e.g., the economic costs derived from using the API, if any, should be reasonable).
- *Robustness to internal error* (e.g., the API should not have bugs in its functioning).
- *User safety* (e.g., the API should neither cause the user legal problems nor compromise personal information and assets).
- *Subjective satisfaction* (e.g., it should be satisfying to work with the API).

Our heuristics and guidelines were integrated into a usability study of an API for a decision support system in the field of sleep medicine. This usability study included activities such as heuristic evaluation and subjective analysis, and yielded the following results:

- With the help of the heuristics and guidelines, we were able to identify 17 main usability problems during the heuristic evaluation (those marked with a “No” answer).
- The majority of these usability issues were resolved after the heuristic evaluation. Only one of these issues identified as “No” in the heuristic analysis (the lack of code samples) was still identified as “No” in the subjective evaluation and it was not fixed due to the early stage of development of the API.
- The questionnaire and subsequent interview with API users helped us to identify areas that we think that were OK after the heuristic evaluation but the API users did not agree. So we obtained “No” answers to questions that were not identified in that way in the heuristic analysis. For example, those questions related with the level of abstraction of the API.
- Some of these usability issues might not have been detected using the previously existing heuristics and guidelines in the literature. More specifically, those related to Efficiency, Robustness, Safety and

Subjective Satisfaction. For example, we detected that the original API was excessively occupying limited resources (ER-1) and the users complained about the API not clearly stating its license of use (SUL-2).

- Our analysis of context of medical sleep applications helped us to highlight which positive or negative results were more relevant to our domain of application. This analysis can be also generalized to similar applications in the same domain.

To conclude, we can say that developing an API is a difficult task, and assessing if an API is usable is therefore also a complicated task. The literature on API usability is very heterogeneous and we consider that the study presented in this paper will contribute to a more global and comprehensive view of the usability of APIs since it is based on a comprehensive taxonomy of usability, which has allowed us to identify aspects of the usability of APIs traditionally neglected by other authors.

Moreover, involving users in the assessment of an API at early stages facilitates its development. Many of the decisions that we have to make are trade-offs, and the opinions of users are fundamental to decide to which side we must tip the balance.

As future work, we are thinking of expanding this study with new users trying to involve them in the development of the API. Also we want to analyse how these heuristics and guidelines work in different environments and tools, not only restricting ourselves to a traditional way of programming but also trying to embrace new ways of programming such as, for example, the development of RESTful APIs for the implementation of Web Services. RESTful APIs are different from traditional APIs because they represent a stateless client/server communication using the HTTP protocol. Since our heuristics were developed for traditional APIs based on object-oriented programming, in which objects represent an encapsulated state, we may have to adapt these heuristics, but always using the taxonomy of usability and the taxonomy of context of use as our working framework.

Also, we are considering adding new usability techniques to our methodology of evaluation of APIs (incorporating quantitative metrics into the heuristics, using different subjective analysis methods, asking users to write actual code with the API under analysis, etc.). An interesting idea would be comparing our methods to other methods published to evaluate API usability and to see the different conclusions that can be obtained.

Acknowledgments

This work has been supported by Xunta de Galicia (Spain) under project GRC2014/035, and Ministerio de Economía y Competitividad (MINECO, Spain), under project TIN2013-40686-P MINECO. Also financial support from the Xunta de Galicia (Centro singular de investigación de Galicia accreditation 2016–2019, REF. ED431G/01) and the European Union (European Regional Development Fund - ERDF), is gratefully acknowledged.

References

[1] D. Alonso-Ríos, A. Vázquez-García, E. Mosqueira-Rey, V. Moret-Bonillo, Usability: a critical analysis and a taxonomy, Int. J. Hum.Comput. Interact. 26 (1) (2009) 53–74.

[2] D. Alonso-Ríos, A. Vázquez-García, E. Mosqueira-Rey, V. Moret-Bonillo, A context-of-use taxonomy for usability studies, Int. J. Hum. Comput. Interact. 26 (10) (2010) 941–970.

[3] D. Alonso-Ríos, E. Mosqueira-Rey, V. Moret-Bonillo, A Taxonomy-based usability study of an intelligent speed adaptation device, Int. J. Hum.Comput. Interact. 30 (7) (2014) 585–603.

[4] D. Álvarez-Estévez, Diagnosis of the Sleep Apnea-Hypopnea Syndrome: a Comprehensive Approach Through an Intelligent System to Support Medical Decision, (PhD Thesis), University of A Coruña, 2012.

[5] D. Álvarez-Estévez, J.M. Fernández-Pastoriza, E. Hernández-Pereira, V. Moret-Bonillo, A method for the automatic analysis of the sleep macrostructure in continuum, Expert Syst. Appl. 40 (5) (2013) 1796–1803.

[6] D. Álvarez-Éstevez, V. Moret-Bonillo, Computer-assisted diagnosis of the sleep apnea-hypopnea syndrome: a review, Sleep Disord. 2015 (2015), http://dx.doi.org/10.1155/2015/237878 Article ID 237878.

[7] American Academy of Sleep Medicine, International Classification of Sleep Disorders—Third Edition (ICSD-3), AASM Resource Library, 2014.

[8] K. Arnold, Programmers are people, too, ACM Queue 3 (5) (2005) 54–59.

[9] K. Beck, Implementation Patterns, Pearson Education, 2007.

[10] R.B. Berry, R. Brooks, C.E. Gamaldo, S.M. Harding, C. Marcus, B. Vaughn, The AASM manual for the scoring of sleep and associated events, Rules, Terminology and Technical Specifications, American Academy of Sleep Medicine, Darien, Illinois, 2016.

[11] M.F. Bertoa, J.M. Troya, A. Vallecillo, Measuring the usability of software components, J. Syst. Softw. 79 (3) (2006) 427–439.

[12] N. Bevan, Extending quality in use to provide a framework for usability measurement, International Conference on Human Centered Design, Berlin Heidelberg, Springer, 2009, July, pp. 13–22.

[13] J. Bloch, (2005). How to design a good API and why it matters. Retrieved September 9, 2016, from http://research.google.com/pubs/archive/32713.pdf.

[14] J. Bloch, Effective Java (2nd Edition) (The Java Series), Prentice Hall PTR, Upper Saddle River, NJ, USA, 2008.

[15] C. Bore, S. Bore, Profiling software API usability for consumer electronics, 2005 *Digest of Technical Papers. International Conference on Consumer Electronics*, 2005, ICCEIEEE, 2005, January, pp. 155–156.

[16] S. Clarke, Measuring API usability, Dr. Dobbs J. 29 (5) (2004) S1–S5.

[17] S. Clarke, Describing and measuring API usability with the cognitive dimensions, Cognitive Dimensions of Notations 10th Anniversary Workshop, 2005, p. 131.

[18] K. Cwalina, B. Abrams, Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .Net Libraries, Pearson Education, 2008.

[19] J.M. Daughtry, U. Farooq, B.A. Myers, J. Stylos, API usability: report on special interest group at CHI, ACM SIGSOFT Softw. Eng. Notes 34 (4) (2009) 27–29.

[20] A. Doucette, On API usability: an analysis and an evaluation tool, CMPT816-Software Engineering, Saskatoon, Saskatchewan, Canada, University of Saskatchewan, 2008.

[21] B. Ellis, J. Stylos, B. Myers, The factory pattern in API design: a usability evaluation, Proceedings of the 29th International Conference on Software Engineering, IEEE Computer Society, 2007, May, pp. 302–312, , http://dx.doi.org/10.1109/ICSE.2007.85.

[22] N. Fenton, J. Bieman, Software Metrics: a Rigorous and Practical Approach, CRC Press, 2015.

[23] A. Fernández-Leal, M. Cabrero-Canosa, E. Mosqueira-Rey, V. Moret-Bonillo, A knowledge model for the development of a framework for hypnogram construction, Knowl. Based Syst. 118 (2017) 140–151 https://doi.org/10.1016/j.knosys.2016.11.016.

[24] D. Glasberg, K. El-Emam, W. Memo, N. Madhavji, Validating object-oriented design metrics on a commercial java application, National Research Council of Canada, 2000.

[25] T.R.G. Green, M. Petre, Usability analysis of visual programming environments: a ‘cognitive dimensions’ framework, J. Vis. Lang. Comput. 7 (2) (1996) 131–174.

[26] T. Grill, O. Polacek, M. Tscheligi, Methods towards API usability: a structural analysis of usability problem categories, International Conference on Human-Centred Software Engineering, Berlin Heidelberg, Springer, 2012, October, pp. 164–180.

[27] M. Henning, API design matters, Queue 5 (4) (2007) 24–36.

[28] IEEE, Institute of Electrical and Electronics Engineers (1992). IEEE Standard for a Software Quality Metrics Methodology, IEEE Std 1061-1992.

[29] ISO 9241-11, Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs), Part 11: Guidance on Usability, International Organization for Standardization, Geneva, 1998.

[30] ISO/IEC 9126, Software Engineering, Product quality, Part 1: Quality Model, International Organization for Standardization, Geneva, 2001.

[31] ISO 15005, Road Vehicles – Ergonomic aspects of Transport Information and Control Systems – Dialogue Management Principles and Compliance Procedures, International Organization for Standardization, Geneva, 2002.

[32] M.Y. Ivory, M.A. Hearst, The state of the art in automating usability evaluation of user interfaces, ACM Comput. Surv. 33 (4) (2001) 470–516.

[33] M. Jacques, (2004, November 21). API Usability: Guidelines to Improve Your Code Ease of Use [Blog post]. Retrieved from http://www.codeproject.com/Articles/8707/API-Usability-Guidelines-to-improve-your-code-ease.

[34] B. Kemp, A. Värri, A.C. Rosa, K.D. Nielsen, J. Gade, A simple format for exchange of digitized polygraphic recordings, Electroencephalogr. Clin. Neurophysiol. 82 (5) (1992, May) 391–393.

[35] B. Kemp, J. Olivan, European data format 'plus' (EDF+), an EDF alike standard format for the exchange of physiological data, Clin. Neurophysiol. 114 (9) (2003, Sep) 1755–1761.

[36] J.R. Lewis, Usability: lessons learned… and yet to be learned, Int. J. Hum. Comput. Interact. 30 (9) (2014) 663–684.

[37] R.C. Martin, Clean code: a Handbook of Agile Software Craftsmanship, Pearson Education, Upper Saddle River, NJ, USA, 2009.

[38] T.J. McCabe, A complexity measure, IEEE Trans. Softw. Eng. (4) (1976) 308–320.

[39] S.G. McLellan, A.W. Roesler, J.T. Tempest, C.I. Spinuzzi, Building more usable APIs, IEEE Softw. 15 (3) (1998) 78–86.

[40] B.A. Myers, J. Stylos, Improving API usability, Commun. ACM 59 (6) (2016) 62–69.

[41] B.A. Myers, A.J. Ko, T.D. LaToza, Y. Yoon, Programmers are users too: human centered methods for improving tools for programming, Computer 49 (7) (2016, July) 44–52.

[42] J. Nielsen, Usability Engineering, Academic Press, Boston, MA, USA, 1993.

[43] J. Nielsen, (1995). 10 Usability Heuristics for User Interface Design. Retrieved from

https://www.nngroup.com/articles/ten-usability-heuristics/.

[44] M. Piccioni, C.A. Furia, B. Meyer, An empirical study of API usability, ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, IEEE, 2013, October, pp. 5–14.

[45] G.M. Rama, A. Kak, Some structural measures of API usability, Softw.: Pract. Exp. 45 (1) (2015) 75–110.

[46] T. Scheller, E. Kühn, Automated measurement of API usability: the API concepts framework, Inf. Softw. Technol. 61 (2015) 145–162.

[47] A. Seffah, M. Donyaee, R.B. Kline, H.K. Padda, Usability measurement and metrics: a consolidated model, Softw. Qual. J. 14 (2) (2006) 159–178.

[48] C.R. de Souza, D.L. Bentolila, Automatic evaluation of API usability using complexity metrics and visualizations, 31st International Conference on Software Engineering (ICSE), IEEE, 2009, May, pp. 299–302 Companion Volume.

[49] J. Stylos, S. Clarke, Usability implications of requiring parameters in objects' constructors, Proceedings of the 29th International Conference on Software Engineering, IEEE Computer Society, 2007, May, pp. 529–539.

[50] J. Stylos, B.A. Myers, The implications of method placement on API learnability, Sixteenth ACM SIGSOFT Symposium on Foundations of Software Engineering (FSE 2008), ACM, 2008, November, pp. 105–112.

[51] J. Tulach, Practical API design: Confessions of a Java Framework Architect, Apress, 2008.

[52] S. Winter, S. Wagner, F. Deissenboeck, A comprehensive model of usability, Engineering Interactive Systems, Springer, Berlin Heidelberg, 2008, pp. 106–122.

[53] M. Zibran, What makes APIs difficult to use, Int. J. Comput. Sci. Netw. Secur. 8 (4) (2008) 255–261.

[54] M.F. Zibran, F.Z. Eishita, C.K. Roy, Useful, but usable? factors affecting the usability of APIs, 2011 *18th Working Conference on Reverse Engineering*, IEEE, 2011, October, pp. 151–155.

Appendix A

EEG: wave patterns

Electroencephalogram (EEG) is the signal that monitors the brain electrical activity, placing electrodes over multiples areas of the scalp, generally in a bipolar setting where one extreme is attached to a specific region and the other to a reference one. It is the most complex of the neurophysiological signals involved in the characterization of sleep as it is non-linear, non-stationary, and has a low signal-to-noise ratio. When studying the EEG, some signal frequencies are important:

- **Delta** is the frequency range up to 4 Hz. It tends to be the highest in amplitude and the slowest waves. It is seen normally in adults in slow-wave sleep. It is also seen normally in babies.
- **Theta** is the frequency range from 4 Hz to 7 Hz. Theta is seen normally in young children. It may be seen in drowsiness or arousal in older children and adults; it can also be seen in meditation.
- **Alpha** is the frequency range from 7 Hz to 13 Hz. It emerges with closing of the eyes and with relaxation, and attenuates with eye opening or mental exertion.
- **Beta** is the frequency range from 14 Hz to about 30 Hz. Beta activity is closely linked to motor behavior and is generally attenuated during active movements. Low-amplitude beta with multiple and varying frequencies is often associated with active, busy or anxious thinking and active concentration.
- **Gamma** is the frequency range approximately 30–100 Hz. Gamma rhythms

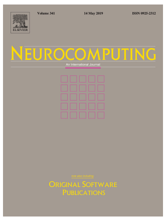
are thought to represent binding of different populations of neurons together into a network for the purpose of carrying out a certain cognitive or motor function.

- **Mu** range is 8–13 Hz and partly overlaps with other frequencies. It reflects the synchronous firing of motor neurons in rest state.

Appendix B

Publications

Included in this thesis

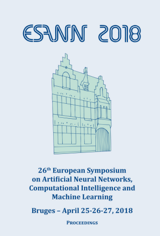


Isaac Fernández-Varela, Elena Hernández-Pereira, Diego Alvarez-Estevez, Vicente Moret-Bonillo. Combining machine learning models for the automatic detection of EEG arousals. Neurocomputing, 268, 100-108, 2017.

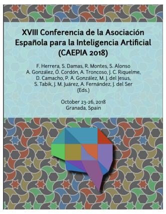
Isaac Fernández-Varela, Diego Alvarez-Estevez, Elena Hernández-Pereira, Vicente Moret-Bonillo. A simple and robust method for the automatic scoring of EEG arousals in polysomnographic recordings. Computers in Biology and Medicine, 87, 77-86, 2017.

Diego Alvarez-Estevez, Isaac Fernández-Varela. Large-scale validation of an automatic EEG arousal detection algorithm using different heterogeneous databases. Sleep Medicine, 57, 6-14, 2019.

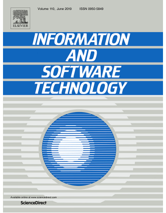
Elena Hernández-Pereira, Isaac Fernández-Varela, Vicente Moret-Bonillo. A Comparison of Performance of Sleep Spindle Classification Methods Using Wavelets. Innovation in Medicine and Healthcare 2016, 60, 61-70, 2016.


ESANN 2018
26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning
Bruges – April 25-26-27, 2018
Proceedings

Isaac Fernández-Varela, Dimitrios Athanasakis, Samuel Parsons, Elena Hernández-Pereira, Vicente Moret-Bonillo. Sleep Staging with Deep Learning : A convolutional model. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 367-372, 2018.


XVIII Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 2018)
October 23-26, 2018
Granada, Spain

Isaac Fernández-Varela, Elena Hernández-Pereira, Diego Alvarez-Estevez, Vicente Moret-Bonillo. A Convolutional Network for the Classification of Sleep Stages. XVIII Conferencia de la Asociacion Española para la Inteligencia Artificial, 1185-1190, 2018.



Eduardo Mosqueira-Rey, David Alonso-Ríos, Vicente Moret-Bonillo, Isaac Fernández-Varela, Diego Alvarez-Estevez. A systematic approach to API usability: Taxonomy-derived criteria and a case study. Information and Software Technology, 97, 46-63, 2018.

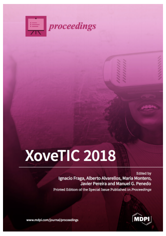
Other publications during the PhD

ESANN 2016
24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning
Bruges – April 27-28-29, 2016
Proceedings

Isaac Fernández-Varela, Elena Hernández-Pereira, Diego Alvarez-Estevez, Vicente Moret-Bonillo. Automatic detection of EEG Arousals. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 235-240, 2016.

ESANN 2017
25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning
Bruges – April 26-27-28, 2017
Proceedings

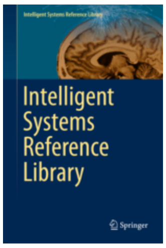
Isaac Fernández-Varela, Diego Alvarez-Estevez, Elena Hernández-Pereira, Vicente Moret-Bonillo. Outlining a simple and robust method for the automatic detection of EEG arousals. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 489-494, 2017.


proceedings
XoveTIC 2018
Edited by
Ignacio Fraga, Alberto Alvarellos, Maria Montero, Javier Pereira and Manuel G. Penedo
Printed Edition of the Special Issue Published in Proceedings
www.mdpi.com/journal/proceedings
MDPI

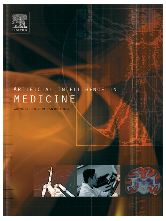
Isaac Fernández-Varela, Elena Hernández-Pereira, Vicente Moret-Bonillo. A Convolutional Network for the Classification of Sleep Stages. XoveTIC, 10-13, 2018.

JENUI
ACTAS
XXIV Jornadas sobre la Enseñanza Universitaria de la Informática
Barcelona, del 4 al 6 de julio de 2018
Organizadas por:
Asociación de Enseñantes Universitarios de la Informática
Estudis d'Informàtica, Multimèdia i Telecomunicació
Universitat Oberta de Catalunya
Editores:
Maria Jesús Marco Galindo
David Bañeres Besora
Josep Maria Marco Simó
jenui2018.uoc.edu

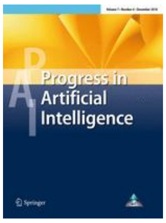
Verónica Bolón-canedo, Amparo Alonso-betanzos, David Alonso-ríos, Isaac Fernández-varela, Daniel Varela. Mejora de la motivación del alumnado mediante la realización de un debate en la materia de Sistemas Inteligentes. Actas de las XXIV Jornadas sobre Enseñanza Universitaria de la Informática, 3, 239-246, 2018.


Intelligent Systems Reference Library
Intelligent Systems Reference Library
Springer

Vicente Moret-Bonillo, Isaac Fernández-Varela, Elena Hernández-Pereira, Diego Alvarez-Estevez, Volker Perlitz. On The Automation of Medical Knowledge and Medical Decision Support Systems. Intelligent Systems Reference Library, 137, 2018.



Diego Alvarez-Estevez, Isaac Fernández-Varela. Dealing with the database variability problem in learning from medical data: an ensemble-based approach using convolutional neural networks and a case of study applied to automatic sleep scoring. Artificial Intelligence in Medicine, 2019. In review



Vicente Moret-Bonillo, Isaac Fernández-Varela, Diego Alvarez-Estevez. Uncertainty in Quantum Rule-based Systems. Progress in Artificial Intelligence, 2019. In review

Appendix C

Resumen

C.1 Introducción

El sueño es un estado de reposo en el que el cerebro no está consciente. Normalmente el cuerpo está tumbado, no se producen movimientos musculares y no se responde a los estímulos externos. Además, su duración es limitada en contraposición con otros estados fisiológicos como el coma o la hibernación. Durante el sueño se disparan diversos mecanismos incluyendo cambios hormonales, procesos de termorregulación o biomecánicos. Aún desconocemos todas las funciones y consecuencias de un sueño reparador, pero sí hemos podido demostrar que es una actividad fundamental para la supervivencia.

Desgraciadamente, las enfermedades relacionadas con el sueño (trastornos del sueño) afectan a una parte importante de la población. Por ejemplo, entre el 30 y el 40% de la población adulta se queja de insomnio y entre un 5 y un 15% de somnolencia diurna. La especialidad médica que diagnostica y trata estos trastornos es la medicina del sueño. Se considera que esta especialidad surge en la segunda mitad del sigo XX, pese a ello no se ofrece como las especializaciones clásicas y, por ejemplo, no hay consenso sobre que material debe tener una unidad del sueño o que entrenamiento debe seguir el especialista.

Los especialistas pueden diagnosticar los trastornos del sueño analizando los datos obtenidos durante un estudio de sueño, que normalmente se realiza en una unidad del sueño. Los datos se analizan para caracterizar la macro estructura (fases de sueño) y la microestructura (eventos como micro despertares o husos de sueño).

Gracias al análisis se pueden diagnosticar los trastornos para, posteriormente, proponer un tratamiento.

La principal dificultad a la hora de realizar este análisis es la gran cantidad de datos que involucra. Antiguamente, el estudio de sueño más habitual (la polisomnografía) generaba, para un paciente y una noche, 500 metros de papel. Además, el acuerdo entre expertos es inferior al 90%. Para la clasificación de fases de sueño, Stepnowsky et al. [5] obtuvieron índices de acuerdo kappa entre 0.46 y 0.89. Para la clasificación de micro eventos es incluso más bajo. El acuerdo publicado para los micro despertares, por ejemplo, está en el intervalo 0.47-0.57 [7, 8].

En esta tesis se presentan algoritmos para la caracterización automática de la macro y microestructura del sueño, utilizando los datos registrado en una polisomnografía. El objetivo es solucionar el cuello de botella actual de las unidades del sueño y mejorar la cohesión y consistencia de los análisis actuales.

Antes de describir los algoritmos desarrollados describiremos el dominio de nuestro problema, centrándonos principalmente en la polisomnografía, en la macro estructura (fases de sueño) y en dos micro eventos: los micro despertares y los husos de sueño. En concreto describiremos dos algoritmos para la detección de micro despertares (incluyendo la validación de uno de ellos en un entorno real), uno para la detección de husos de sueño y dos para la clasificación de las etapas de sueño. Además, incluímos un último artículo con un caso de estudio para la construcción de una API que facilite el uso de nuestras propuestas. Este resumen termina con las conclusiones obtenidas realizando este trabajo y una breve descripción de los trabajos propuestos para continuar.

C.1.1 Polisomnografía

Los estudios del sueño son importantes porque permiten caracterizar patrones para, comparándolos con patrones normales, diagnosticar los trastornos del sueño. Aunque existen diversos tipos como la actigrafía o el test múltiple de latencia de sueño, el más habitual e importante es la polisomnografía (PSG). Este estudio consiste en la grabación de múltiples señales neurofisiológicas durante el sueño de un paciente, colocando sensores en su cuerpo. El número de señales y sensores es variable, seleccionándose en función de las sospechas del médico que solicita el estudio. Es útil para diagnosticar varios tipos de trastornos incluyendo narcolepsia, hipersomnias, desorden de movimiento límbico periódico, desorden del comportamiento REM, parasomnias y apnea del sueño. Además, es útil para descartar otros y para detectar episodios transitorios como despertares, paseos nocturnos, bruxismo o terrores nocturnos.

El montaje habitual del PSG registra tres tipos de señales: neumológicas, neurofisiológicas y contextuales.

Señales neumológicas

Son aquellas relacionadas con la actividad respiratoria, la saturación de oxígeno en sangre y el flujo de aire, siendo éstas dos últimas las más comunes.

Señales neurofisiológicas

Son las señales directamente relacionadas con el sueño y, por tanto, las que utilizaremos en nuestros algoritmos. Las más importantes son el electrooculograma (EOG), el electromiograma (EMG) y el electroencefalograma (EEG).

- **Electrooculograma (EOG):** es la señal que registra el movimiento de los ojos. Es importante para distinguir distintos patrones de movimientos oculares que ocurren durante el sueño.
- **Electromiograma (EMG):** es la señal que registra la actividad muscular. Normalmente se registra la actividad de la barbilla porque refleja cambios en el estado del sueño y la actividad tibial para controlar los movimientos de las piernas.
- **Electroencefalograma (EEG):** es la señal que monitoriza la actividad cerebral. Sin lugar a dudas la más compleja de las señales relacionadas con el sueño al ser no lineal, no estacionaria y tener una mala relación señal ruido.

Señales contextuales

Son las señales que no están relacionadas directamente con el sueño del paciente como los ruidos en la habitación o su postura corporal.

C.1.2 Estructura del sueño

Para caracterizar el sueño de un paciente es necesario definir tanto su macro estructura como detectar los distintos eventos pertenecientes a la microestructura. Por eso en esta tesis proponemos algoritmos para la clasificación de las fases de sueño (macro estructura) y para la detección de dos de los eventos más importantes: micro despertares y husos de sueño.

C.1.3 Fases de sueño

Loomis et al. [14] fueron los primeros en observar que el sueño no es un estado homogéneo y que nuestra actividad cerebral pasa por distintas fases. En 1954, Aserinsky and Kleitman observaron una fase particular del sueño caracterizada por movimientos rápidos de los ojos (REM) [13] y como se repetía cíclicamente el patrón de fase REM y fase no REM [23]. Ellos mismos dividieron la fase no REM en 4, desde sueño ligero a profundo, basándose en las diferencias encontradas en la actividad cerebral. Fueron Rechtschaffen and Kales (R&K) [17] los que estandarizaron la definición de las fases de sueño basándose en las señales de EOG, EMG y EMG. El manual que publicaron incluía parámetros, técnicas y patrones comunes para clasificar las fases de sueño a partir de una polisomnografía. De hecho, el manual R&K fue el estándar de facto hasta el 2007. Era necesario, sin embargo, un estándar real que permitiese comparar distintos estudios y mejorar la reproducibilidad de los resultados de otras unidades de sueño. Siguiendo el manual R&K, que sugería definir una fase de sueño para cada ventana de 30 segundos (epoch) y el análisis estructurado del sueño, en 2007 la Academia Americana de Medicina del Sueño (AASM) publica una guía estandarizada. Las reglas publicadas por la AASM siguen en su mayoría las propuestas de R&K, incluyendo mejoras propiciadas por los últimos avances y técnicas. Los cambios más significativos son la fusión de la fase 3 y 4 en una única fase de sueño profundo y la distinción de las fases para pacientes pediátricos. El manual de la AASM define un total de cinco fases: Despierto (W), Movimientos oculares rápidos (REM), Fase 1 (N1), Fase 2 (N2) y Fase 3 (N3).

- **Despierto (W):** es la fase que representa el estado despierto, desde que estamos totalmente alerta a la somnolencia inicial del sueño.
- **Fase 1 (N1):** es la fase de sueño más ligero, en la que percibimos la mayoría

de estímulos a nuestro alrededor.

- **Fase 2 (N2):** es la fase en la que dejamos de responder a estímulos y el sueño empieza a ser reparador, aunque no por completo.
- **Fase 3 (N3):** es la fase de sueño profundo. Si nos despertamos en esta fase lo habitual es sentirse desorientado.
- **Fase REM:** es la fase durante la que soñamos.

C.1.4 Micro despertares

Uno de los eventos de la microestructura del sueño. Se define como un cambio abrupto de la frecuencia de la señal de EEG incluyendo la banda *alpha*, *theta* y frecuencias superiores a 16 Hz (pero no la banda *spindle*) que dura por lo menos 3 segundos y viene precedido por 10 segundos de sueño estable. Son una respuesta en forma de alerta que se produce durante el sueño sin llegar a suponer un despertar completo, pero que suelen provocar el cambio a una fase de sueño más ligera. Son, sin duda, un indicador excelente de la calidad del sueño.

C.1.5 Husos de sueño

Los husos de sueño se definen como un tren de ondas con frecuencias entre 11 y 16 Hz (típicamente entre 12 y 14) que dura por lo menos medio segundo y que alcanza la máxima amplitud hacia la mitad del evento. Son un claro indicio de que el sueño está en fase 2 y uno de los pocos eventos que se pueden detectar en la señal de EEG que únicamente están relacionados con el sueño.

C.2 Detección de micro despertares

El primer método que presentamos sigue la aproximación clásica de clasificar un vector de características obtenidas sobre las señales. El segundo intenta superar algunas de las limitaciones del anterior, reconociendo patrones sobre la señal.

C.2.1 Combining machine learning models for the automatic detection of EEG arousals

Este trabajo intenta encontrar el mejor conjunto de características para la detección de micro despertares y cómo combinar múltiples clasificadores individuales. Primero, encontramos los segmentos de las señales de EEG y EMG en los que se podría detectar un micro despertar, buscando cambios en las señales. En el caso del EEG el cambio se busca estudiando la potencia de la señal mientras que para el EMG estudiamos la amplitud. Si dentro de un mismo epoch encontramos cambios para el EEG y para el EMG consideramos que es un epoch relevante. Para cada epoch relevante construimos un vector de características, incluyendo las magnitudes comparadas antes, los parámetros de Hjorth [36] y la fase de sueño del epoch. Utilizando un base de datos de 20 PSG del *Sleep Heart Health* Study [37] construimos dos *datasets* balanceados para entrenamiento y test. Entrenamos seis clasificadores: discriminante lineal [38], *support vector machine*, *neural network* [39], *classification tree*, *k-nearest neigbor* y *naive bayes* [41]. Los cuatro clasificadores con mejor rendimiento los combinamos usando dos aproximaciones. La primera sigue el modelo de factores de incertidumbre de Shortliffe and Buchanan [42] y la segunda una combinación lineal.

Finalmente, comprobamos el rendimiento de las combinaciones propuestas utilizando un nuevo *dataset* de 26 registros polisomnográficos. Nuestra propuesta mejora los resultados de los clasificadores y de *ensembles* conocidos como *random forest* [43] y *k-NN* [44]. Con el modelo de factores de incertidumbre obtuvimos una sensitividad de 0.78, especificidad de 0.89 y un error de 0.12. Con la combinación lineal una sensitividad de 0.81, una especificidad de 0.88 y un error de 0.13.

C.2.2 A simple and robust method for the automatic scoring of EEG arousals in polysomnographic recordings

El objetivo de este trabajo es simplificar el proceso de detección de micro despertares. Así conseguimos un algoritmo que se puede integrar fácilmente con software ya existente y, además, una detección de micro despertares que es fácil de explicar. Como en el algoritmo anterior, empezamos buscando cambios abruptos en la frecuencia de la señal analizando la potencia de una ventana y comparándola con las anteriores. Cada uno de los cambios encontrados los estudiamos para encontrar patrones comunes de los micro despertares.

Reconocemos tres posibles patrones. El primero se basa en la potencia de la señal de EEG, el segundo se basa en la amplitud de la señal de EEG y el último en la amplitud de la señal de EMG. Posteriormente descartamos los patrones reconocidos si son falsos positivos. Para ello comprobamos que la longitud sea adecuada, que vengan precedido por diez segundos de sueño estable, que no sea un huso del sueño y que si sucede durante REM venga acompañado por actividad en la señal de EMG. El método se comprobó utilizando un *dataset* de 22 registros de PSG obtenido en la unidad de sueño del Haaglanden Medisch Centrum (HMC) en La Haya, Países Bajos. La precisión obtenida fue de 0.86, con un acuerdo kappa casi perfecto (0.78).

C.2.3 Large-scale validation of an automatic EEG arousal detection algorithm using different heterogeneous databases

En este trabajo mejoramos nuestro algoritmo anterior para para mejorar su capacidad de generalización. Lo adaptamos para se pueda utilizar con montajes diferentes (conjuntos de señales o frecuencia de muestreo entre otros) y actualizamos algunos valores para mejorar la capacidad de detectar micro despertares en la banda *alpha*.

El objetivo del artículo es evaluar el algoritmo utilizando *datasets* de distintas fuentes, por lo que incluimos registros del SHHS y del HMC. Para evaluarlo utilizamos dos aproximaciones. Por un lado, comparar el *Arousal Index* para obtener coeficientes de correlación entre el algoritmo y la referencia clínica. Por otro, medimos la confiabilidad entre métodos para la detección (humano vs máquina). Respecto al *Arousal Index* los resultados permiten aceptar que la mediana de las diferencias entre métodos es 0 si asumimos una desviación de 0.3. En cuanto a la confiabilidad, los resultados permiten asegurar que el algoritmo se comporta como un experto más, con índices de acuerdo máquina contra experto similares a los obtenidos cuando comparamos dos expertos entre sí.

C.3 Detección de husos de sueño

Para la detección de husos de sueño incluimos un trabajo que, como en el caso de los micro despertares, utiliza clasificación sobre un vector de características extraído de la señal.

C.3.1 A comparison of performance of sleep spindle classification method using wavelets

En este trabajo encontramos husos de sueño clasificando un vector de características. En este caso, el *dataset* [57] contiene segmentos de señal correctamente anotados. Las características que extraemos de cada segmento son los coeficientes de descomposición de la señal utilizando una función *wavelet* [58]. En la evaluación del trabajo comparamos distintas familias de *wavelets* y varios clasificadores. Utilizando validación cruzada la mejor exactitud se consiguió con una *wavelet* biortognal de orden 1.5 y un *random forest*. La mejor senstividad también se consiguió con dicha familia y orden de *wavelet* pero utilizando como clasificador un *proximal support vector machine*.

C.4 Clasificación de fases de sueño

La clasificación automática de fases de sueño es, sin duda, el problema que más veces se ha tratado de resolver en este campo. Sin embargo, lo más común es encontrar trabajos que realizan la clasificación de un vector de características. El problema es que estas características suelen estar escogidas por el investigador, basándose en su conocimiento del dominio y los datos de los que dispone. Indudablemente estas características no son imparciales y, por ello, las soluciones propuestas no suelen generalizar bien.

Los trabajos que presentamos utilizan *deep learning* para mitigar este efecto. Este tipo de métodos aprenden por sí solos qué características de la señal son necesarias para la clasificación, evitando nuestra propia imparcialidad.

C.4.1 Sleep staging with deep learning: a convolutional method

Para nuestra primera aproximación a la clasificación de fases de sueño utilizando *deep learning* simplificamos un poco el problema, fusionando las fases N1 y N2 en una única fase de sueño ligero. Utilizamos una red neuronal que recibe como entrada un *epoch* (incluyendo dos derivaciones de EEG, ambos EOG y EMG) y proporciona como salida la probabilidad de pertenencia a cada fase de sueño, seleccionando la más alta como resultado.

Utilizamos un *dataset* de entrenamiento para encontrar la mejor arquitectura de red neuronal y los mejores valores de hiperparámetros, resultando en un total de tres capas convolucionales. Al clasificar un *dataset* de test con 40 registros, la precisión obtenida para cada fase de sueño estaba entre 0.89 y 0.96, y el acuerdo F1 entre 0.85 y 0.96.

C.4.2 A convolutional method for the classification of sleep stages

Este trabajo es una mejora del anterior. En primer lugar, eliminamos las simplificaciones, clasificando las cinco fases del sueño. Además, mejoramos la selección de hiperparámetros y de la arquitectura de la red.

Para la selección de la arquitectura y de hiperparámetros utilizamos un estimador (*Tree-structure Parzen Estimator*) [81]. Es un modelo de optimización secuencial que entrena modelos y en función de los resultados selecciona nuevos valores para el siguiente entrenamiento. Siguiendo este método entrenamos 50 modelos distintos y seleccionamos los cinco mejores para construir un *ensemble*. Con el *ensemble* clasificamos 500 registros y obtuvimos una precisión media de 0.78, una sensitividad media de 0.75 y un acuerdo medio (*F1 score*) de 0.76.

C.5 Construyendo una API para medicina del sueño

Una de las razones para que los algoritmos que analizan la macro y microestructura del sueño no salgan del ámbito académico es la dificultad de integrarlos con software existente. Intentado mitigar este problema presentamos un trabajo de usabilidad de APIs utilizando como caso de estudio la construcción de una API para utilizar los algoritmos descritos.

C.5.1 A systematic approach to API usability: taxonomy-derived criteria an a case study

En este trabajo se presentan guías y heurísticas para la usabilidad de una API que sintetizan estudios de usabilidad previos y cubren puntos inexistentes. Utilizamos las guías y heurísticas para mejorar una API para medicina del sueño, corrigiendo errores que no hubiésemos encontrado siguiendo los trabajos de usabilidad previos.

Para construir la API analizamos el contexto de uso, consiguiendo una primera versión sobre la que hicimos una evaluación heurística. Los resultados se utilizaron para corregir los errores de la primera versión de la API. La nueva versión se entregó a posibles usuarios para su evaluación subjetiva. La evaluación subjetiva se realiza con un cuestionario que también le proporcionamos, para recoger las respuestas que cubren los puntos de nuestra heurística. Gracias a la evaluación subjetiva conseguimos nuevos puntos de mejora para la siguiente iteración.

C.6 Conclusiones y trabajo futuro

La medicina del sueño se podría beneficiar de algoritmos que analicen la estructura del sueño. Si pudiésemos automatizar esta tarea, evitando el tiempo que consume en la actualidad, los médicos podrían centrarse en el diagnóstico y tratamiento. Cambiaría por completo el cuello de botella de las unidades del sueño.

Para caracterizar la microestructura del sueño presentamos algoritmos que detectan micro despertares y husos de sueño. El primero de los métodos para las micro despertares clasifica un vector de características, incluyendo la fase de sueño y los parámetros de Hjorth porque mejoran la capacidad de detección. Podemos incluso mejorar la sensitividad y especificidad de métodos ya publicados utilizando un *ensemble* que combina clasificadores individuales siguiendo el modelo de factores de incertidumbre de Shortliffe y Buchanan. El segundo método para la detección de micro despertares los encuentra analizando patrones en la señal. Aunque el objetivo de este método era que fuese sencillo, el acuerdo alcanzado con el experto es mejor que en trabajos de otros autores. Además, con pequeñas modificaciones mejoramos su capacidad de generalización. Al ejecutarlo en un entorno real (una unidad de sueño en un hospital de La Haya, Países Bajos) se comporta como uno más de los expertos que trabajan en dicho entorno. En cuanto a la detección de husos de sueño demostramos que la caracterización de la señal con wavelets, aunque no se haya utilizado con anterioridad en este problema, permite los mismos resultados que otras caracterizaciones más habituales.

Para clasificar las fases de sueño utilizamos una aproximación diferente, evitando nuestra propia imparcialidad. En este caso proponemos el uso de redes convolucionales que puedan aprender por si mismas qué características son relevantes para clasificar las fases de sueño. Tras una primera aproximación en la que simplificamos el problema (clasificando solo ciertas fases), mejoramos la solución para resolver el problema real. Configurando la red con algoritmos automáticos la clasificación que hacemos para ciertas clases (W y N3) es la mejor (comparada con otros trabajos anteriores) mientras que para el resto de las clases es muy parecida a la que obtienen otros autores.

Finalmente, intentado facilitar la integración de nuestros algoritmos construimos una API. Nuestro diseño inicial lo mejoramos utilizando guías y heurísticas. Después entrevistamos a posibles usuarios que evaluaban el API de una manera subjetiva. Aunque la mayor parte de problemas los habíamos corregido gracias a las guías y heurísticas, conseguimos información relevante para una próxima mejora.

C.6.1 Trabajo futuro

El trabajo futuro propuesto encaja en dos líneas principales. Por un lado, explicabilidad y por otro la mejora de los algoritmos propuestos. Los algoritmos de inteligencia artificial utilizados actúan como una caja negra. Es difícil explicar cómo llegan a la solución y eso complica su análisis y mejora. Además, incrementa la desconfianza del experto, limitando su aplicación en entornos reales. En cuanto a la mejora de los algoritmos las propuestas siguen las líneas marcadas por nuestros últimos trabajos. Así, el objetivo será utilizar *deep learning* para la detección de micro eventos y la mejora de la clasificación de fases de sueño con redes más compleja que incorporen memoria al sistema.

Bibliography

[1] Daniel J. Buysse, Lan Yu, Douglas E. Moul, Anne Germain, Angela Stover, Nathan E. Dodds, Kelly L. Johnston, Melissa A. Shablesky-Cade, and Paul A. Pilkonis. Development and Validation of Patient-Reported Outcome Measures for Sleep Disturbance and Sleep-Related Impairments. *Sleep*, 33(6):781–792, jun 2010. ISSN 1550-9109.

[2] Nancy A Collop. Conundrums in sleep medicine. *Chest*, 115(3):607, 1999.

[3] Ruth M. Benca. Diagnosis and Treatment of Chronic Insomnia: A Review. *Psychiatric Services*, 56(3):332–343, mar 2005. ISSN 1075-2730.

[4] Ángel Fernández-Leal, Mariano Cabrero-Canosa, Eduardo Mosqueira-Rey, and Vicente Moret-Bonillo. A knowledge model for the development of a framework for hypnogram construction. *Knowledge-Based Systems*, 118:140–151, 2017. ISSN 09507051.

[5] Carl Stepnowsky, Daniel Levendowski, Djordje Popovic, Indu Ayappa, and David M Rapoport. Scoring accuracy of automated sleep staging from a bipolar electroocular recording compared to manual scoring by multiple raters. *Sleep medicine*, 14(11):1199–207, nov 2013. ISSN 1878-5506.

[6] Ying Wang, Kenneth A Loparo, Monica R Kelly, and Richard F Kaplan. Evaluation of an automated single-channel sleep staging algorithm. *Nature and science of sleep*, 7:101–11, 2015. ISSN 1179-1608.

[7] Michael J. Drinnan, Alan Murray, Clive J. Griffiths, and G. John Gibson. Interobserver Variability in Recognizing Arousal in Respiratory Sleep Disorders. *American Journal of Respiratory and Critical Care Medicine*, 158(2):358–362, aug 1998. ISSN 1073-449X.

[8] Warren R. Ruehland, Thomas J. Churchward, Linda M. Schachter, Tristia Lakey, Natalie Tarquinio, Fergal J. O'Donoghue, Maree Barnes, and Peter D. Rochford. Polysomnography using abbreviated signal montages: impact on sleep and cortical arousal scoring. *Sleep Medicine*, 16(1):173–180, jan 2015. ISSN 1389-9457.

[9] Tarek Asaad. Sleep in Ancient Egypt. In *Sleep Medicine*, pages 13–19. Springer New York, New York, NY, 2015.

[10] Henri Piéron. Le problème physiologique du sommeil. In *Le probléme physiologique du sommeil*, page 520. Masson, 1913.

[11] René Legendre and Henri Piéron. *Du développement, au cours de l'insomnie expérimentale, de propriétés hypnotoxiques des humeurs en relation avec le besoin croissant de sommeil.* 1911.

[12] Clifford B Saper, Thomas C Chou, and Thomas E Scammell. The sleep switch: hypothalamic control of sleep and wakefulness. *Trends in Neurosciences*, 24(12):726–731, dec 2001. ISSN 0166-2236.

[13] Eugene Aserinsky, Nathaniel Kleitman, and Others. Regularly occurring periods of eye motility, and concomitant phenomena, during sleep. *Science*, 118(3062):273–274, 1953.

[14] A. L. Loomis, E. N. Harvey, and G. A. Hobart. Cerebral states during sleep, as studied by human brain potentials. *Journal of Experimental Psychology*, 21(2):127–144, 1937. ISSN 0022-1015.

[15] Nathaniel Kleitman. Sleep and Wakefulness as Alternating Phases in the Cycle of Existence. *Journal of the American Medical Association*, 113(23), 1939.

[16] M Jouvet, F Michel, and J Courjon. On a stage of rapid cerebral electrical activity in the course of physiological sleep. *Comptes rendus des seances de la Societe de biologie et de ses filiales*, 153:1024–1028, 1959.

[17] A Rechtschaffen and A Kales. A manual of standardized terminology, technique and scoring system for sleep stages of human sleep. *Brain Information Service, Los Angeles*, 1968.

[18] A Rechtschaffen, M A Gilliland, B M Bergmann, and J B Winter. Physiological correlates of prolonged sleep deprivation in rats. *Science (New York, N.Y.)*, 221(4606):182–4, jul 1983. ISSN 0036-8075.

[19] D L Bliwise, N G Bliwise, M Partinen, A M Pursley, and W C Dement. Sleep apnea and mortality in an aged cohort. *American journal of public health*, 78(5):544–7, may 1988. ISSN 0090-0036.

[20] Meir Kryger, Thomas Roth, and William C. Dement. *Principles and practice of sleep medicine.* Elsevier, 6 edition, 2017. ISBN 9780323242882.

[21] Murray W. Johns. A New Method for Measuring Daytime Sleepiness: The Epworth Sleepiness Scale. *Sleep*, 14(6):540–545, nov 1991. ISSN 0161-8105.

[22] Richard B. Berry, Rita Brooks, Charlene Gamaldo, Susan M. Harding, Robin M. Lloyd, Stuart F. Quan, Matthew T. Troester, and Bradley V. Vaughn. AASM Scoring Manual Updates for 2017 (Version 2.4). *Journal of Clinical Sleep Medicine*, 2017. ISSN 1550-9389.

[23] William Dement and Nathaniel Kleitman. Cyclic variations in EEG during sleep and their relation to eye movements, body motility, and dreaming. *Electroencephalography and Clinical Neurophysiology*, 9(4):673–690, nov 1957. ISSN 0013-4694.

[24] Luigi De Gennaro and Michele Ferrara. Sleep spindles: an overview. *Sleep Medicine Reviews*, 7(5):423–440, oct 2003. ISSN 1087-0792.

[25] Hans Berger. Über das Elektrenkephalogramm des Menschen. *Archiv für Psychiatrie und Nervenkrankheiten*, 87(1):527–570, dec 1929. ISSN 0003-9373.

[26] A. L. Loomis, E. N. Harvey, and G. Hobart. Potential rhythms of the cerebral cortex during sleep. *Science*, 81(2111):597–598, jun 1935.

[27] Giora Pillar, Amir Bar, Arie Shlitner, Robert Schnall, Jacob Shefy, and Peretz Lavie. Autonomic arousal index: an automated detection based on peripheral arterial tonometry. *Sleep*, 25(5):543–549, 2002. ISSN 0161-8105.

[28] Stefan Telser, Martin Staudacher, Yvonne Ploner, Anton Amann, Hartmann Hinterhuber, and Monika Ritsch-Marte. Can One Detect Sleep Stage Transitions for On-Line Sleep Scoring by Monitoring the Heart Rate Variability?. Sind Schlafstadienwechsel durch eine on-line Analyse der Herzschlagvariabilitat erkennbar? *Somnologie*, 8(2):33–41, may 2004. ISSN 1432-9123.

[29] Pedro Gouveia, Ricardo Oliveira, a C da Rosa, and Agostinho Rosa. Sleep apnea related micro-arousal detection with EEG analysis. In *Proc. 7th Portuguese Conf. on Biomed. Eng.*, 2003.

[30] S.P. Cho, J. Lee, H.D. Park, and K.J. Lee. Detection of Arousals in Patients with Respiratory Sleep Disorders Using a Single Channel EEG. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, volume 3, pages 2733–2735. IEEE, 2005. ISBN 0-7803-8741-4.

[31] Rajeev Agarwal. Automatic Detection of Micro-Arousals. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, volume 2, pages 1158–1161. IEEE, 2005. ISBN 0-7803-8741-4.

[32] F De Carli, L Nobili, P Gelcich, and F Ferrillo. A method for the automatic detection of arousals during sleep. *Sleep*, 22(5):561–572, 1999. ISSN 0161-8105.

[33] Ursula Malinowska, P J Durka, K J Blinowska, Waldemar Szelenberger, and Andrzej Wakarow. Micro- and macrostructure of sleep EEG. *IEEE Engineering in Medicine and Biology Magazine*, 25(4):26–31, jul 2006. ISSN 0739-5175.

[34] O.R. Pacheco and F. Vaz. Integrated system for analysis and automatic classification of sleep EEG. In *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol.20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No.98CH36286)*, pages 2062–2065. IEEE, 1998. ISBN 0-7803-5164-9.

[35] Sobhan Salari Shahrbabaki, Chamila Dissanayaka, Chanakya Reddy Patti, and Dean Cvetkovic. Automatic detection of sleep arousal events from polysomnographic biosignals. In *2015 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, oct 2015. ISBN 978-1-4799-7234-0.

[36] Bo Hjorth. EEG analysis based on time domain properties. *Electroencephalography and Clinical Neurophysiology*, 29(3):306–310, sep 1970. ISSN 0013-4694.

[37] Stuart F. Quan, Barbara V. Howard, Conrad Iber, James P. Kiley, F. Javier Nieto, George T. O'Connor, David M. Rapoport, Susan Redline, John Robbins, Jonathan M. Samet, and Patricia W. Wahl. The Sleep Heart Health Study: Design, Rationale, and Methods. *Sleep*, 20(12):1077–1085, dec 1997. ISSN 1550-9109.

[38] R. A. FISHER. THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS. *Annals of Eugenics*, 7(2):179–188, sep 1936. ISSN 20501420.

[39] Jose C Principe, Neil Euliano, and C Lefebvre. Neural systems: Fundamentals through Simulations. *Online Book of NeuroSolutions v3. 0*, 1999.

[40] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, mar 1986. ISSN 0885-6125.

[41] T Mitchell, B Buchanan, G DeJong, T Dietterich, P Rosenbloom, and A Waibel. Machine Learning. *Annual Review of Computer Science*, 4(1):417–433, jun 1990. ISSN 8756-7016.

[42] Edward H. Shortliffe and Bruce G. Buchanan. A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23(3-4):351–379, apr 1975. ISSN 0025-5564.

[43] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 08856125.

[44] C. Domeniconi and B. Yan. Nearest neighbor ensemble. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, pages 228–231 Vol.1. IEEE, 2004. ISBN 0-7695-2128-2.

[45] T. Sugi, F. Kawana, and M. Nakamura. Automatic EEG arousal detection for sleep apnea syndrome. *Biomedical Signal Processing and Control*, 4(4):329–337, 2009. ISSN 17468094.

[46] Oren Shmiel, Tomer Shmiel, Yaron Dagan, and Mina Teicher. Data mining techniques for detection of sleep arousals. *Journal of Neuroscience Methods*, 179(2):331–337, 2009. ISSN 01650270.

[47] Diego Alvarez-Estevez and Vicente Moret-Bonillo. Identification of electroencephalographic arousals in multichannel sleep recordings. *IEEE Transactions on Biomedical Engineering*, 58(1):54–63, 2011. ISSN 00189294.

[48] Diego Alvarez-Estevez, Jose M Fernandez-Pastoriza, Elena Hernández-Pereira, and Vicente Moret-Bonillo. On the continuous evaluation of the macrostructure of sleep. *Frontiers in Artificial Intelligence and Applications*, 243:189–198, 2012. ISSN 09226389.

[49] Isaac Fernández-Varela, Elena Hernández-Pereira, Diego Alvarez-Estevez, and Vicente Moret-Bonillo. Combining machine learning models for the automatic detection of EEG arousals. *Neurocomputing*, 268:100–108, dec 2017. ISSN 0925-2312.

[50] Patrick E. Shrout and Joseph L. Fleiss. Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin*, 86(2):420–428, 1979. ISSN 1939-1455.

[51] S. Devuyst, T. Dutoit, J. F. Didier, F. Meers, E. Stanus, P. Stenuit, and M. Kerkhofs. Automatic sleep spindle detection in patients with sleep disorders. *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, pages 3883–3886, 2006. ISSN 05891019.

[52] Errikos M. Ventouras, Efstratia a. Monoyiou, Periklis Y. Ktonas, Thomas Paparrigopoulos, Dimitris G. Dikeos, Nikos K. Uzunoglu, and Constantin R. Soldatos. Sleep spindle detection using artificial neural networks trained with filtered time-domain EEG: A feasibility study. *Computer Methods and Programs in Biomedicine*, 78(3):191–207, 2005. ISSN 01692607.

[53] Syed Anas Imtiaz, Siavash Saremi-Yarahmadi, and Esther Rodriguez-Villegas. Automatic detection of sleep spindles using Teager energy and spectral edge frequency. In *2013 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 262–265. IEEE, oct 2013. ISBN 978-1-4799-1471-5.

[54] D. Gorur. *Automated Detection of Sleep Spindles.* PhD thesis, The Middle East Technical University, 2003.

[55] Nurettin Acır and Cüneyt Güzeliş. Automatic recognition of sleep spindles in EEG by using artificial neural networks. *Expert Systems with Applications*, 27(3):451–458, oct 2004. ISSN 0957-4174.

[56] Salih Güneş, Mehmet Dursun, Kemal Polat, and Sebnem Yosunkaya. Sleep spindles recognition system based on time and frequency domain features. *Expert Systems with Applications*, 38(3):2455–2461, mar 2011. ISSN 0957-4174.

[57] Devuyst, Stéphanie, Dutoit, Thierry, Stenuit, Patricia, Kerkhofs, and Myriam. Automatic Sleep Spindles Detection – Overview and Development of a Standard Proposal Assessment Method. In *Annual international conference of the IEEE engineering in medicine and biology society.* IEEE, 2011. ISBN 9781424441228.

[58] Raghuveer M. Rao and Ajit S. Bopardikar. *Wavelet transforms : introduction to theory and applications.* Addison-Wesley, 1998. ISBN 0201634635.

[59] Olvi L. Mangasarian and Edward W. Wild. Proximal support vector machine classifiers. In *PROCEEDINGS KDD-2001: KNOWLEDGE DISCOVERY AND DATA MINING*, pages 77–86, 2001.

[60] Luay Fraiwan, Khaldon Lweesy, Natheer Khasawneh, Heinrich Wenz, and Hartmut Dickhaus. Automated sleep stage identification system based on time–frequency analysis of a single EEG channel and random forest classifier. *Computer Methods and Programs in Biomedicine*, 108(1):10–19, oct 2012. ISSN 01692607.

[61] Jian Liang, Rui Lu, Changshui Zhang, and Fei Wang. Predicting Seizures from Electroencephalography Recordings: A Knowledge Transfer Strategy. In *Proceedings - 2016 IEEE International Conference on Healthcare Informatics, ICHI 2016*, pages 184–191. IEEE, oct 2016. ISBN 9781509061174.

[62] Guohun Zhu, Yan Li, and Peng Paul Wen. Analysis and Classification of Sleep Stages Based on Difference Visibility Graphs From a Single-Channel EEG Signal. *IEEE Journal of Biomedical and Health Informatics*, 18(6):1813–1821, nov 2014.

[63] Ahnaf Rashik Hassan and Mohammed Imamul Hassan Bhuiyan. A decision support system for automatic sleep staging from EEG signals using tunable Q-factor wavelet transform and spectral features. *Journal of Neuroscience Methods*, 271:107–118, sep 2016. ISSN 0165-0270.

[64] Ahnaf Rashik Hassan and Mohammed Imamul Hassan Bhuiyan. Computer-aided sleep staging using Complete Ensemble Empirical Mode Decomposition with Adaptive Noise and bootstrap aggregating. *Biomedical Signal Processing and Control*, 24:1–10, feb 2016. ISSN 1746-8094.

[65] Rajeev Sharma, Ram Bilas Pachori, and Abhay Upadhyay. Automatic sleep stages classification based on iterative filtering of electroencephalogram signals. *Neural Computing and Applications*, 28(10):2959–2978, oct 2017. ISSN 0941-0643.

[66] B. Koley and D. Dey. An ensemble system for automatic sleep stage classification using single channel EEG signal. *Computers in Biology and Medicine*, 42(12):1186–1195, 2012. ISSN 00104825.

[67] Tarek Lajnef, Sahbi Chaibi, Perrine Ruby, Pierre-Emmanuel Aguera, Jean-Baptiste Eichenlaub, Mounir Samet, Abdennaceur Kachouri, and Karim Jerbi. Learning machines and sleeping brains: Automatic sleep stage classification using decision-tree multi-class support vector machines. *Journal of Neuroscience Methods*, 250:94–105, jul 2015. ISSN 0165-0270.

[68] Chih-Sheng Huang, Chin-Teng Chun-Ling Lin, Li-Wei Ko, Shen-Yi Liu, Tung-Ping Su, and Chin-Teng Chun-Ling Lin. Knowledge-based identification of sleep stages based on two forehead electroencephalogram channels. *Frontiers in Neuroscience*, 8:263, sep 2014. ISSN 1662-453X.

[69] Salih Günes, Kemal Polat, and Sebnem Yosunkaya. Efficient sleep stage recognition system based on EEG signal using k-means clustering based feature weighting. *Expert Systems with Applications*, 37(12):7922–7928, dec 2010. ISSN 0957-4174.

[70] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, feb 2017. ISSN 0028-0836.

[71] Varun Gulshan, Lily Peng, Marc Coram, Martin C. Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, Ramasamy Kim, Rajiv Raman, Philip C. Nelson, Jessica L. Mega, and Dale R. Webster. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA*, 316(22):2402, dec 2016. ISSN 0098-7484.

[72] Martin Längkvist, Lars Karlsson, Amy Loutfi, and Amy Loutfi. Sleep Stage Classification Using Unsupervised Feature Learning. *Advances in Artificial Neural Systems*, 2012:1–9, 2012. ISSN 1687-7594.

[73] Orestis Tsinalis, Paul M. Matthews, Yike Guo, and Stefanos Zafeiriou. Automatic Sleep Stage Scoring with Single-Channel EEG Using Convolutional Neural Networks, oct 2016, 1610.01683.

[74] Orestis Tsinalis, Paul M. Matthews, and Yike Guo. Automatic Sleep Stage Scoring Using Time-Frequency Analysis and Stacked Sparse Autoencoders. *Annals of Biomedical Engineering*, 44(5):1587–1597, may 2016. ISSN 0090-6964.

[75] Akara Supratak, Hao Dong, Chao Wu, and Yike Guo. DeepSleepNet: A Model for Automatic Sleep Stage Scoring Based on Raw Single-Channel EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(11):1998–2008, nov 2017. ISSN 1534-4320.

[76] Siddharth Biswal, Joshua Kulas, Haoqi Sun, Balaji Goparaju, M Brandon Westover, Matt T Bianchi, and Jimeng Sun. SLEEPNET: Automated Sleep Staging System via Deep Learning, jul 2017, 1707.08262.

[77] Arnaud Sors, Stéphane Bonnet, Sébastien Mirek, Laurent Vercueil, and Jean-François Payen. A convolutional neural network for sleep stage scoring from raw single-channel EEG. *Biomedical Signal Processing and Control*, 42:107–114, apr 2018. ISSN 17468094.

[78] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. dec 2014, 1412.6980.

[79] Vinod Nair and Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on Machine Learning*, (3): 807–814, 2010, 1111.6189v1. ISSN 1935-8237.

[80] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015, 1502.03167. ISSN 0717-6163.

[81] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for Hyper-Parameter Optimization. In *NIPS*, 2011.

[82] D. Alonso-Ríos, A. Vázquez-García, E. Mosqueira-Rey, and V. Moret-Bonillo. Usability: A Critical Analysis and a Taxonomy. *International Journal of Human-Computer Interaction*, 26(1):53–74, dec 2009. ISSN 1044-7318.