NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTAMENT OF PHYSICS

RODRIGO DUARTE BRAGA

Bachelor of Science in Biomedical Engineering

MACHINE LEARNING ALGORITHMS DEVELOPMENT FOR SLEEP CYCLES DETECTION AND GENERAL PHYSICAL ACTIVITY BASED ON BIOSIGNALS

MASTER IN BIOMEDICAL ENGINEERING

NOVA University Lisbon
September, 2022

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTAMENT
OF PHYSICS

MACHINE LEARNING ALGORITHMS DEVELOPMENT FOR SLEEP CYCLES DETECTION AND GENERAL PHYSICAL ACTIVITY BASED ON BIOSIGNALS

RODRIGO DUARTE BRAGA

Bachelor of Science in Biomedical Engineering

Adviser: Prof. Dr. Hugo Filipe Silveira Gamboa
Associate Professor, FCT-NOVA

Examination Committee:

Chair: Prof. Dr. Susana Isabel dos Santos Silva Sério Venceslau
Assistant Professor, FCT-NOVA

Rapporteurs: Dr. Miquel Alfaras Espinàs
Postdoctoral researcher, University of Jaume I

Adviser: Prof. Dr. Hugo Filipe Silveira Gamboa
Associate Professor, FCT-NOVA

MASTER IN BIOMEDICAL ENGINEERING

NOVA University Lisbon
September, 2022

Machine learning algorithms development for sleep cycles detection and general physical activity based on biosignals

Acknowledgements

At the end of this phase of my academic life, I have to look back and thank everyone that supported me throughout this journey.

First of all, this challenging project would not have been possible without Professor Hugo Gamboa's availability and receptiveness, who, in the role of adviser of this thesis, both presented me with the opportunity of learning through developing a topic I'm deeply interested in, but also allowed me to work with creative independence.

I am also extremely grateful to Engineer Daniel Osório who tirelessly guided me during this dissertation, and was always available with helpful suggestions whenever problems or questions arose.

I would like to extend my sincere thanks to the people at PLUX, for the warm reception and atmosphere, which made me feel welcome there and allowed me to start this work on the right foot.

Additionally, I am grateful all the support and help that was made available by my friends that, despite their own difficulties and goals, helped lighten the load and stress that I felt through it all.

Lastly, I would be remiss in not mentioning my family for always believing and supporting me in whichever way they could regardless of how tough times were, especially my mother, father, and brother.

Abstract

In this work, machine learning algorithms for automatic sleep cycles detection were developed. The features were selected based on the AASM manual, which is considered the gold standard for human technicians. These include features such as saturation of peripheral oxygen or others related to heart rate variation. As normally, the sleep phases naturally differ in frequency, to balance the classes within the dataset, we either oversampled the least common sleep stages or undersampled the most common, allowing for a less skewed performance favouring the most represented stages, while simultaneously improving worst-stage classification.

For training the models we used MESA, a database containing 2056 full overnight unattended polysomnographies from a group of 2237 participants. With the goal of developing an algorithm that would only require a PPG device to be able to accurately predict sleep stages and quality, the main channels used from this dataset were SpO2 and PPG.

Employing several popular Python libraries used for the development of machine learning and deep learning algorithms, we exhaustively explored the optimisation of the manifold parameters and hyperparameters conditioning both the training and architecture of these models in order for them to better fit our purposes.

As a result of these strategies, we were able to develop a neural network model (*Multilayer perceptron)* with 80.50% accuracy, 0.7586 Cohen's kappa, and 77.38% F1-score, for five sleep stages. The performance of our algorithm does not seem to be correlated with sleep quality or the number of transitional epochs in each recording, suggesting uniform performance regardless of the presence of sleep disorders.

To test its performance in a different real-world scenario we compared the classifications attributed by a popular sleep stage classification android app, which collected information using a smartwatch, and our algorithm, using signals obtained from a device developed by PLUX. These algorithms displayed a strong level of agreement (90.96% agreement, 0.8663 Cohen's kappa).

Keywords : Deep Learning; Machine Learning; Wearable; Photoplethysmography; Sleep Stages; Heart Rate Variation.

Resumo

Neste trabalho, foram desenvolvidos algoritmos de aprendizagem de máquinas para a detecção automática de ciclos de sono. Os sinais específicos captados durante a extração de características foram selecionados com base no manual AASM, que é considerado o padrão-ouro para técnicos. Estas incluem características como a saturação do oxigénio periférico ou outras relacionadas com a variação do ritmo cardíaco. A fim de equilibrar a frequência das classes dentro do conjunto de dados, ora se fez a sobreamostragem das fases menos comuns do sono, ora se fez a subamostragem das mais comuns, permitindo um desempenho menos enviesado em favor das fases mais representadas e, simultaneamente, melhorando a classificação das fases com pior desempenho.

Para o treino dos modelos criados, utilizámos MESA, uma base de dados contendo 2056 polissonografias completas, feitas durante a noite e sem vigilância, de um grupo de 2237 participantes.

Do conjunto de dados escolhido, os principais canais utilizados foram SpO2 e PPG, com o objetivo de desenvolver um algoritmo que apenas exigiria um dispositivo PPG para poder prever com precisão as fases e a qualidade do sono.

Utilizando várias bibliotecas populares de Python para o desenvolvimento de algoritmos de aprendizagem de máquinas e de aprendizagem profunda, explorámos exaustivamente a optimização dos múltiplos parâmetros e hiperparâmetros que tanto condicionam a formação como a arquitetura destes modelos, de modo a que se ajustem melhor aos nossos propósitos.

Como resultado disto, fomos capazes de desenvolver um modelo de rede neural (*Multilayer perceptron*) com 80.50% de precisão, 0.7586 kappa de Cohen e F1-score de 77.38%, para cinco fases de sono. O desempenho do nosso algoritmo não parece estar correlacionado com a qualidade do sono ou o número de épocas de transição em cada gravação, sugerindo um desempenho uniforme independentemente da presença de distúrbios do sono.

Para testar o seu desempenho num cenário de mundo real diferente, comparámos as classificações atribuídas por uma aplicação Android de classificação de fases do sono popular, através da recolha de informação por um smartwatch, e o nosso algoritmo, utilizando sinais obtidos a partir de um dispositivo desenvolvido pela PLUX. Estes algoritmos demonstraram um forte nível de concordância (90.96% de concordância, 0.8663 kappa de Cohen).

Palavras-chave: Aprendizagem profunda; Aprendizagem de máquinas; Wearable; Fotopletismografia; Estágios de Sono; Variação do Ritmo Cardíaco.

Contents

List of Figures

List of Tables

Abbreviations

AASM – American Academy of Sleep Medicine

CVD – Cardiovascular disease

EEG - Electroencephalogram

ECG – Electrocardiogram

EMG – Electromyogram

EOG – Electrooculogram

HR – Heart rate

ML – Machine learning

MESA – Multi-Ethnic Study of Atherosclerosis

NN – Neural network

NREM – Non-rapid eye movement

OSA – Obstructive sleep apnea

PPG – Photoplethysmography

PSG – Polysomnography

REM – Rapid eye movement

Sp02 – saturation of peripheral oxygen

1. Introduction

1.1. Context and Motivation

Sleep's impact on mood and health is so widely recognized by medical researchers that such understanding disseminated among average people in recent years [2]. While newer studies strengthen the suspected link between inadequate sleep and a wide range of infirmities [2], the general population is not very conscious of their sleep quality. This can lead to not giving enough importance to proper rest, as people are not aware of their accumulating sleep deficits or the toll these deficits take on their conscious cognitive functions [3, 4]. Simultaneously, there is also a tendency for people to overestimate their sleep periods in self-reports [5].

As a result, there is much interest in having proper means of studying sleep, given its importance and how difficult it is to accurately diagnose sleep disorders, considering how individuals are affected by sleep loss, and their ability to recover from said sleep loss, varies significantly depending on individual phenotypic traits [6].

The discovery of the brain's electrical activity was the main contributor responsible for the development of the field of sleep medicine in the second half of the 20th century [7]. The examination of the electroencephalogram (EEG) patterns that occur during sleep lead to the current division of the sleep period into different stages, thus creating the basis of sleep medicine and the study of human sleep [3]. Therefore, we came to understand, among other particulars, that sleep is much more restorative to both waking cognition and health when it occurs accordingly to our circadian clock and goes through the appropriate physiological sequences (Figure 1.1) [3].

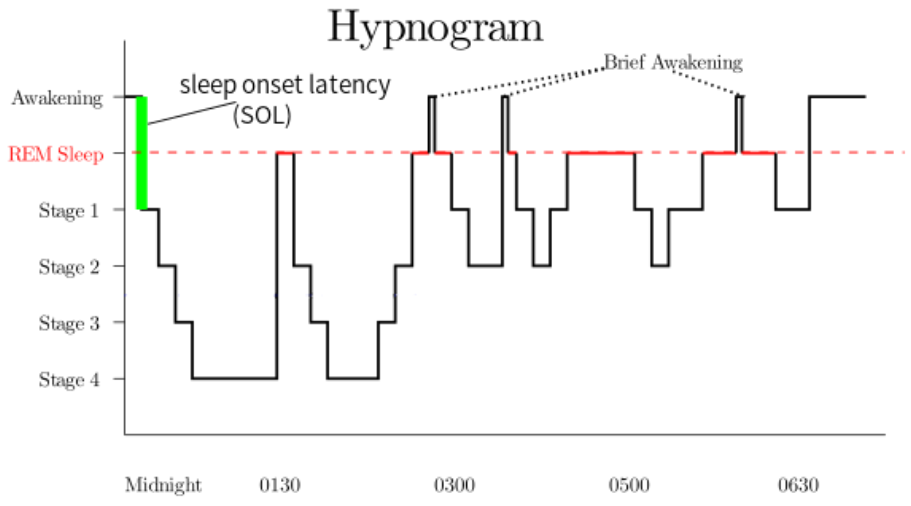

Figure 1.1 - Hypnogram detailing 6.5 hours of sleep. Adapted from [8].

This is to say that, due to the way that sleep is structured into distinct stages, where each one has a certain set of characteristics and its own physiological role, the exclusive measurement of the amount of time slept is not enough for the quality of sleep to be determined.

As such, sleep quality depends not only on total time slept but on many other factors such as fragmentation, amount of time spent in each sleep stage, and how the sleep cycles are structured.

There are two types of sleep, non-rapid eye movement (NREM) sleep and rapid eye movement (REM) sleep [9]. Complementarily, sleep is divided into 5 stages: wake, N1, N2, N3, and REM (with some studies and standards further dividing N3 into N3 and N4), where N stands for NREM sleep and represents a progression of relative sleep depth. Most of sleep's duration (c.a. 75%) is spent in the NREM stages, with a typical night's sleep consisting of 4 to 5 sleep cycles, in the following order: N1, N2, N3, REM [10] (averaging 90 minutes for each cycle [11]), with the majority of time spent in the N2 stage [12]. As mentioned previously, each of these stages has unique characteristics that allow us to distinguish them between themselves, ranging from variations of brain wave patterns to eye movements or muscle tone.

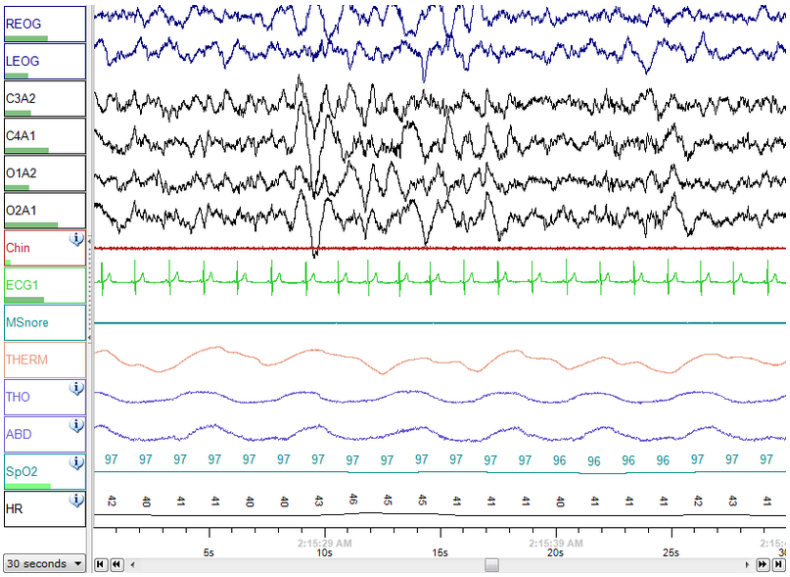

Figure 1.2 – Screenshot of a person in stage N3 sleep, showcasing the signals obtained from different channels over a 30-second window. Retrieved from [13].

Although sleep cycles and stages were uncovered through the use of EEG and the study of the electrical patterns of brain activity [14, 15], other methodologies can be applied to study them. For example, the metabolic differences that arise during sleep or the variation of the different degrees of activation of either the parasympathetic or sympathetic nervous systems [16].

The detection of these metabolic changes can be achieved through the measurement of temperature, heart, or breathing rate. On the other hand, although these characteristics are also correlated to the functioning of parasympathetic or sympathetic nervous systems, determining the degree of activation of either of them is not so straightforward, especially when not utilising EEG, as is the case when using photoplethysmography (PPG) or electrocardiography (ECG). Some studies have used heart rate variability (HRV) to obtain information about the autonomic nervous systems and to predict sleep stages [12, 17, 18].

In general, there is a decline in heart rate (HR) and sympathetic-nerve activity during NREM sleep, with this being more noticeable in deeper stages and reflecting a decrease in the level and variability of arterial pressure [19]. On the other hand, parasympathetic activity tends to increase as sleep transitions from wakefulness to NREM sleep [20]. Simultaneously, this trend is somewhat reversed during REM sleep where, instead, there is an increase in sympathetic activity, even when compared to wakefulness [19].

Currently, polysomnography (PSG) is the most common technique used to study sleep disorders, being able to record brain activity (EEG), heart (ECG), breathing rate, muscle activity (electromyography, EMG), and electro-ocular activity (electrooculography, EOG) [21], with some exams also recording respiratory effort, blood oxygen saturation and performing video analysis [22]. It has, however, the issue of being expensive and inconvenient [23]. This type of exam is normally performed in a clinic, which raises the issue of negative bias, as people may behave differently than normal when they know they are being monitored [24].

Furthermore, there is the matter of longitudinal data. Laboratory PSGs are usually a single-night snapshot, whereas sleep is a dynamic process that is affected by the existence and intensity of many other factors that vary from day to day, such as exercise, caffeine intake, diet, and stress, among others. As a result of the natural variation and the interactions that can result from these, comes the need for improved “real-world” measurements and statistics, derived from large datasets that would allow correlations to be established between different factors and quality of sleep. From a practical point of view, for any one individual, such data can only be acquired outside of a clinical environment. This can be achieved, for example, through self-monitoring, which traditionally implies methods such as sleep diaries that, as mentioned previously, pose some problems due to their unreliability. As a consequence, due to the vaster amount of information that is possible to capture reliably this way, there has been an increasing interest in assisting self-monitoring through the use of wearable devices.

Wearable sleep-trackers, for instance, in the form of wristbands, smartwatches or headbands, are low-cost devices capable of measuring several biosignals, such as heart rate, temperature, sweat levels (through skin conductance measurements), in addition to motion [25]. This data can then be used by wearables devices to infer information about certain behaviours, like sleep, or to diagnose disorders, such as obstructive sleep apnea (OSA) [26].

Besides the convenience and affordability of these devices, there are also other advantages that they present, such as their user-friendliness and data accessibility, with many of these devices having some sort of cloud-based platform used for storage and integration of said data. In this way, they allow the acquisition of an unprecedented amount of information about sleep and other behaviours or health parameters [27], throughout extended periods in peoples' natural environments, with minimal inconvenience to users, who simply have to wear the device, lessening the burden of specialized technicians that would examine the already processed data provided by these devices. Another advantage is that the novelty and widespread use of devices with health monitoring capabilities (like smartwatches and smartphones) have also contributed to the dissemination of the importance of sleep in our modern society [3], while additionally familiarizing people with this kind of technology.

However, despite the seemingly many different initial advantages of these devices when compared to other more traditional exams, there are some drawbacks. To begin with, these devices have a battery that needs to be charged frequently, which when combined with the possibility of the novelty of such devices fading to the individual, might lead to lower persistent use of the devices over time than would otherwise be expected. Moreover, unlike in PSG where after the data is collected it is examined by specialists, data extracted by wearables tends to be directly processed and interpreted by an algorithm inbuilt into the device. This poses the issue of producing an algorithm that is applicable across a diverse population, given a wide variety of sleep problems and other comorbidities that might influence the aspect of sleep physiology measured by said device, and as such, affect the scoring reliability of the algorithm and, thus, of the device itself [23].

Not surprisingly, the growing interest in this area means that more validation studies are being made. However, this effort is still unable to keep up with the rate at which these devices are developed and introduced to the market [25]. When this is combined with the fact that consumer wearables are commercial devices, which use proprietary algorithms that are sometimes subject to change over time, makes it difficult to understand exactly how accurate these devices are.

This is why the amount of data available is so important, where, despite the existence of so many other databases, researchers still argue for the need of popular wearables already being employed in this field to share their raw data access. This possibility would not only mitigate some validation concerns, but also serve as a foundation for the development of future algorithms and devices. This is also pertinent since there are many different factors, unrelated to the disorders or other problems that might influence the quality of sleep of an individual, that can influence the data collected depending on the population examined [23, 28].

To add to the problem of validation, although PSG is considered the gold standard for sleep assessment, and as such is used as a comparison to home sleep monitors, it is still imprecise, being manually scored by experienced technicians, who sometimes disagree on their decisions [29, 30]. Finally, while theoretically possible, posterior manual analysis of the stored data is cost prohibitive and time-consuming, as there is a massive amount of data produced by these devices, making this kind of processing and validation that much more difficult. All in all, this makes for a lack of available data about many of these devices' validity, accuracy, and reliability.

As such, taking the above-mentioned information into account, there is interest in the development of algorithms that, with the ability to be paired with wearable devices, would be able to automatically and accurately classify sleep stages with a similar degree of accuracy as the current gold-standard for this area, in order to bypass the associated costs and time expenditures, as well as other issues mentioned previously. Should this be done, a focus on ensuring that the algorithm performs well across a wide population, and that its structure, inner workings, and behavior are carefully documented would be critical. Ideally, such an algorithm should also strive to be as simple as possible (both in terms of signals used and model complexity).

1.2. Objectives

The main objective of this work is the development of a machine learning (ML) algorithm that allows us to detect and classify sleep cycles through the use of data sourced from wearables. Because of this, there is also a focus on minimizing the number of signals and features used, in order to reduce the energy spent and storage space occupied in these devices, while additionally attempting to decrease the time spent processing.

Accordingly, several steps were accomplished during this dissertation:

- Choosing which signals to utilise;
- Selecting the databases to use to train the different models;

- Pre-processing the data and selecting the features later extracted;
- Creation and optimization of the creation process of the developed models;
- Data acquisition, to obtain new data samples;
- Use of the new data acquired to compare the performance with commercially available algorithms;

1.3. Thesis Structure

This work is divided into five chapters. This introductory chapter identifies both the relevancy of the field of study and some of the problems with the technologies currently used in it, which justify further study of this area. The viability of other technologies and the likelihood of their replacement of the gold standard are also evaluated. A description of this thesis' main objectives and structure is also done.

In the second chapter, a review of the literature and an up-to-date description of the state-of-the-art for the relevant technologies involved in this work are done.

In the third chapter, information about the databases chosen to be used in this work is detailed, as well as the methodology used during the development of the thesis in terms of signal pre-processing, feature selection and extraction, ML algorithm development, and evaluation of the performance of the developed models.

In Chapter 4, “Results and Discussion”, we display results, describe the influence of each of the choices made during model development on their performance, and present the characteristics of the developed algorithms.

Finally, in the fifth chapter, after showing the performance of the classifiers, we compare it with those obtained in other studies. Some areas for improvement and additional functionalities for future work or studies are proposed, with the level of fulfilment of the objectives submitted earlier in the thesis being evaluated.

2. State of the Art

According to Pavlova and Latreille [31], sleep disorders are common in modern society but, despite the treatment for some being difficult, most can be easily managed with adequate interventions, as long as they are properly diagnosed.

As previously stated in Sub-Section 1.1., PSG is considered a diagnostic reference tool for sleeping problems, with it traditionally being required to diagnose these disorders [32], mainly due to the amount of information that can be gathered in it.

For most of the past, sleep was thought to be a passive state; it was only in the middle of the 20th century that scientists examined sleep from a physiological perspective. This was only possible due to a deepening in the understanding of both the form and nature of the cells that compose the nervous system as well as the discovery of the electrical activity of the brain (Figure 2.1).

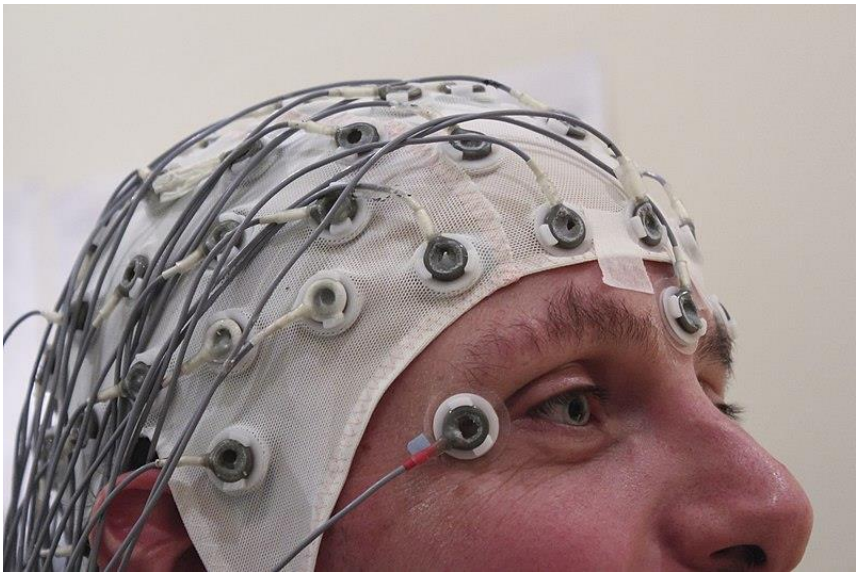

Figure 2.1 - A cap holds electrodes in place while recording an EEG. Retrieved from [33].

Cortical electrical activity in humans was first recorded, from the scalp, in 1924, resulting in the creation of EEG [34]. Shortly after this, most of the major elements of sleep wave patterns were described in a series of experiences [35], with several papers being published between 1935 and 1939 [36–40] describing the principal features that now make up non-REM sleep. This included the division of sleep being into five different stages (A, B, C, D, and E), being arranged in order of appearance and resistance to change by external disturbances. These experiments gave rise to improvements in the methods used to study sleep, starting with advancements in the recording of EEG (through the use of amplifiers and both high and low pass filters), the discovery of specific brain regions that lead to the creation of better records of certain EEG waveforms (ushering greater

importance of certain EEG channels) and, finally, the addition of channels that record other physiological measurements, like heart rate, respiration or temperature [41]. In due course, this effort culminated in the addition of more channels to detect information such as movements during sleep [6], eye movements (initially through direct visualization, eventually through EOG), and muscle potentials (through EMG). This occurred in conjunction with fundamental changes in the way this information was gathered, such as the move towards recording continuously throughout the entirety of a night (as opposed to intermittent sampling of sleep during the night or short sleep recordings). In 1967, the first consensus-based guidelines for staging and scoring sleep, called R & K or Rechtschaffen and Kales system, was developed [34]. These guidelines went through different iterations, with the advent of modern digital equipment eventually leading to the creation of the American Academy of Sleep Medicine (AASM) manual in 2007 [42]. This is a continuously evolving resource, with the latest version (2.6) being released January 2020 [43].

Currently, in most cases, PSG consists of the recording of at least 4 channels (corresponding to EEG, EMG, and two EOG channels) which are then manually scored by experts for the purpose of defining the different sleep stages [44, 45]. The specialised equipment used in PSG and the experts necessary to process the output of this technology are the principal cause for them being costly and time-consuming (scoring an eight-hour-long PSG may require up to two hours of work [28]), often being reflected in long waiting lists [46]. As a result of this, recent studies have focused on improving PSG scoring methods, both in terms of accuracy (through the additional use of other exams [47]), and labour or time required (with a shift from manual scoring to automatic [48]), optimizing the amount of data extracted so as to reduce processing time and storage space [49]. These developments also bring the prospect of a reduced amount of PSG recordings needed to reach a therapeutic decision in some patients via a fully automated analysis [50].

Nevertheless, automatic scoring for PSG is not yet widely used in sleep centres due to high inter-scorer variability and low inter-scorer agreement [51], caused by the scoring rules employed, which allow for some subjective interpretation of the collected data. Another question raised by these rules is related to the division of sleep in 30-second long epochs and considering sleep stages as distinct entities, while a more accurate representation of sleep would be as a gradual transition from one stage to another [28].

A possible strategy to solve issues related to the limited or delayed capacity to perform PSG could be portable monitoring, using a selected number of bio-parameters, which has

been proposed as an alternative in order to shorten the diagnosis time and the beginning of effective treatment for some sleep disorders like OSA [52].

Another increasingly popular alternative is the use of wearable devices [53]. A complete wearable device is composed of two essential parts, hardware and software. Hardware typically involves the selection of the sensors and their characterization, communication (both to the decision-making subsystem and, potentially, to other devices), and noise signal processing. Software, on the other hand, beyond signal processing, is also usually involved in making decisions based on the acquired signals. Recent technical developments made the use of wearable medical devices feasible for real-time monitoring and diagnosis of medical problems, with their growing adoption in healthcare further accelerated due to the availability of cheaper components and advancements in wireless communication technology [54]. Despite this, however, these devices still face challenges in terms of robustness, accuracy, precision, and reliability [55].

Home sleep monitoring devices vary greatly by which and how many different biosignals they monitor, with some devices measuring only one type of biosignals (for example, wrist actigraphy (Figure 2.2) for the detection of insomnia or circadian rhythm disorders [56]), while others monitor many simultaneously for the purpose of reaching a more informed decision.

Figure 2.2 – Example of a smartwatch, capable of wrist actigraphy. Retrieved from [57].

In order to reduce the discomfort caused by these devices (to improve patient compliance), the size of the device, power consumption and storage space necessary, material cost, and time required for analysis or processing (all of which are major design considerations in wearable systems [58]), there has been an effort to record only targeted, specific signals, with recent studies attempting to identify which are the optimal signals to

be used as inputs for automatic sleep stage detection algorithms or sleep disorder detection [49].

Other studies choose to instead focus on the discussion of how these devices acquire data, as well as their objective. For instance, these devices can be classified as wearable (like the devices mentioned in Sub-Section 1.1., wristbands, smartwatches, or headbands), or non-wearable (such as smart mattresses, under-the-mattress sensors, or under-the-sheet sensors), with some being used for sleep stage classification, sleep posture monitoring, sleep disorder detection, vital signs monitoring [59], or even to study circadian rhythm patterns [26]. This type of specialization allows these devices to be as efficient as possible in their desired function, which inadvertently has the possible drawback of making their validation more complex as explained previously.

Another potential downside is that, in such a quickly developing industry, the validation effort may lag behind the release and update of new devices [25], which might differ in the sensors, biosignals, and algorithms used, with both the algorithms and raw data not usually being shared publicly. As a consequence, determining their accuracy and validating them may be laborious [27, 60]. This is especially true when it is taken into account that PSG, despite being considered the golden standard for these devices, has its flaws in this respect [23].

Nevertheless, it is still possible to give an overview of the current devices that have been validated by comparison to PSG or that have already been adopted by individuals for this express purpose. For example, Fitbit Charge 2, which records wrist activity through accelerometers and pulses through PPG, has been validated against portable home PSG, where it was determined to be able to provide reasonably accurate mean values of sleep and HR estimates, should it follow careful data processing [61]. One other device is the Heally Recording System which, through the combination of embedded sensors and electrodes in a shirt that measures respiratory and cardiac physiology, monitors sleep based on autonomic signals. It exhibited accuracy at approximately 80% agreement with manual scoring, which is similar to accuracies obtained through actigraphy [22], considered an appropriate method for the assessment of sleep in patients with certain sleep disorders [62].

As mentioned previously, all these different wearable devices or automatic methodologies for PSG scoring require software or, more specifically, an algorithm to evaluate their input and output results, with its accuracy directly influencing how reliable the device itself is. Some of these first algorithms were elaborated through the use of discriminant analysis techniques [63], currently, there are more advanced techniques to develop these models.

One such technique is machine learning. Considered a subset of artificial intelligence, ML creates dynamic algorithms capable of data-driven decisions, in contrast to computer programs that follow static programming instructions [64, 65]. These algorithms have the ability to automatically improve their performance at some tasks through experience, however, this requires ground truth in the form of accurately curated data sources from carefully constructed subject trials with a properly distributed subject population of significant size [54]. Normally, such requirements would pose a challenge to the training of these algorithms, however, wearables' low cost, ease of use, and unobtrusiveness make widespread, longitudinal studies feasible [66]. This means that it is relatively easier to produce a significant amount of raw data, ready to be used to train algorithms as needed, allowing them to be quickly developed, improved, and overall a more suitable option for model development than before.

In regard to ML itself, there are differences in how these algorithms are produced, with some recent studies delving into which are the optimal ways to perform their training with the data available [49] or testing previously created algorithms with different datasets in order to minimize bias [67].

Other studies, which rely on either ML or deep learning, have successfully developed algorithms for sleep stage prediction. For example, Tsinalis et al. (2020) managed to obtain sleep stage-specific characteristics with an average accuracy of 86% based on EEG data [68], while Yildirim et al. (2019), developed and applied a 19-layer 1D convolutional neural network model to EEG and EOG signals, achieved the highest classification accuracies for 5 of its 6 sleep classes as over 91% [48]. This suggests that the development of similar fully automatic recognition systems could serve as a suitable replacement for manual inspection of PSG signals, particularly for large-scale studies.

There are many considerations for the development of ML algorithms for this purpose, however. To begin with, even analogous problems or otherwise identical questions but with differences in data, tend to have distinct best-case solutions in terms of the ML algorithm approaches [69, 70]. For instance, in the case of deep learning, these differences can manifest themselves in terms of learning methods, types of artificial neural networks (NN), number of layers or neurons in the case of these NNs, activation functions, optimizers for their compilation or the consideration of different optimal metrics that are monitored during training, among many other variations of parameters and hyperparameters.

The optimization of these parameters is pivotal for the development of algorithms with the best possible performance for a specific dataset and problem. Too simple of a model and it will not be able to appropriately learn from the data, underfitting to it, and having a

low performance even for data it has been trained on (high bias). Too complex of a model (in the case no precautions are taken) and it will learn from the noise in the dataset, perhaps displaying high accuracy for the training dataset (due to overfitting), but a considerably lower performance for the test dataset (high variance). This is a delicate balance that is necessary to take into account for the development of models for a particular purpose.

3. Methodology

The main goal of this work was the development of an algorithm that takes biosignals recorded by sensors and converts them into useful information about sleep cycles and the quality of sleep. To do this, it was necessary to be able to discern between the different stages of sleep and correlate certain biosignals with sleep quality. Initially, the algorithm was trained through the use of a publicly available online database, selected from among others such as the NCH Sleep DataBank [71], or the Sleep Heart Health Study [72]. Afterwards, we acquired real-world data, used these models to classify it, and compared the results with other publicly available algorithms.

Accordingly, the first task was the selection of a few variables from specific databases including raw PPG signal, HR, and saturation of peripheral oxygen (SpO2). Following that, it was necessary to apply signal processing techniques, like filtering, to be able to properly extract the information used to train the algorithms. The structure and training process of these algorithms were refined over several iterations until a satisfactory result was achieved. Afterwards, data acquisition protocols were developed for the acquisition of relevant signals and, finally, after this, we proceeded with the evaluation of the effectiveness of the system through the use of real-world data, comparing the results from our model with the ones obtained through the use of a publicly available algorithm.

In order to build the code developed in this thesis to analyse and process the dataset, as well as build the ML models, Python was used through the code editor Spyder. Several different libraries were used, including *BeautifulSoup4*, *Pandas*, *NumPy*, *scikit-learn*, *Tensorflow*, and *hrvanalysis*.

3.1. Chosen Dataset

There are many factors that might influence the quality of sleep of an individual, and, as such, affect the data collected depending on the population examined. For this reason, the present work chose to initially prioritize the training of our algorithm resorting to carefully chosen databases, before testing it with real-world information obtained through the use of devices developed by PLUX. After a comparison and evaluation among these databases, one was selected based on its size, sensor quality and quantity, detail of the scoring, and how recently collected was the data.

The set of PSG recordings[1] used in this thesis was obtained from the *Multi-Ethnic Study of Atherosclerosis* (MESA) [73–75]. This dataset was designed to study the characteristics

[1] We requested authorisation to use this dataset on the 24th of March 2022, with access being granted on the 31st of March the same year.

of subclinical cardiovascular disease (CVD) and the risk factors that predict progression to clinically overt CVD or progression of the subclinical disease. With the objective of understanding how variations in sleep and sleep disorders vary across gender and ethnic groups and relate to measures of subclinical atherosclerosis, a sleep exam was conducted between 2010-2012, with 2237 participants, which included a full overnight unattended polysomnography, these being the exams utilised for this work (Table 3.1).

Table 3.1- Dataset demographics of the MESA database (adapted from [76]).

Characteristics	Value
Number of PSGs	2056
Number of Patients	2237
Age	(Years)
Mean	69.6
Median	69.0
Standard deviation	± 9.2
Minimum	54.0
Maximum	95.0
Gender	
Female	1198
Male	1039
Race/ethnicity	
White, Caucasian	830
Chinese American	265
Black, African-American	616
Hispanic	526

As can be seen in Table 3.1, this sleep study's polysomnography documentation contains 2056 PSG recordings, with the information pertaining to each PSG being split into two separate files.

In this regard, files in the XML format contain annotations corresponding to the PSG recordings, that is, information regarding the type of events (respiratory, sleep stages, among others), description of the events (hypopnea or SpO2 desaturation, for example), time of the start of the events and, finally, the duration of said events. The events relating to stages of sleep are collapsed at the end of the file, with each sleep stage there being similarly defined by stage, duration, and when each stage started. Using these events,

hypnograms, comprised of information concerning the classification of the patient's sleep stages over time, were assembled for each XML file[2].

These sleep stages are scored based on the AASM guidelines [42], using an epoch-by-epoch approach with 30 seconds duration, and each epoch being assigned a single sleep stage score. These epochs are scored into stages W, N1, N2, N3, and R (respectively identified as 0, 1, 2, 3, or 5 in the XML files), corresponding to AASM's wake stage (wakefulness), stage N1, stage N2, both stage N3 and N4 simultaneously, and REM, respectively.

The analysis of the XML files allows us to differentiate the proportion of sleep stages among each other, which is very useful information to later fine-tune the creation process of our algorithms (Figure 3.1).

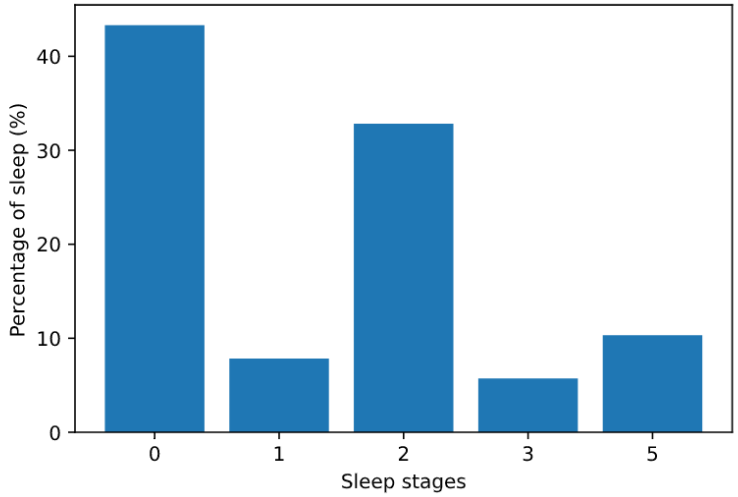

Figure 3.1 - Graphical representation of the relative incidence for each sleep stage in the MESA database.

As expected, the most prevalent sleep stage, from REM and NREM sleep, is stage 2, representing 32.82% of all intervals in the dataset. The overall most common stage in these files is stage 0, representing 43.30%, with a significant part of the counts of this stage being from before the start of sleep and after the end of it. After that, the number of stages 1, 3, and 5 are similar, with 5 being the most common out of these at 10.31%, followed by stage 1 at 7.83%, and, finally, stage 3 is the least common in the database, placed at only 5.73%.

Conversely, the files in the EDF format contain the recorded signals[3]. Each recording makes use of 20 sensors to produce 27 signals. Among these signals are the 3 that were used from this dataset, namely "HR", "Pleth" (which is the PPG recording), and "SpO2",

[2] This was done through the use of the BeautifulSoup4 Python library to read these files.
[3] Which were read by resorting to the mne Python library.

which is extracted through an oximeter. These 3 signals were originally sampled at 1 Hz, except for the “Pleth” signal, which was sampled at 256 Hz.

3.2. Signal Pre-processing

The information in the XML files was used to segment the signals in the EDF files into different 30-second long intervals, and assign a sleep stage for supervised ML.

The relevant features from these intervals were subsequently saved in the CSV files used to later train models. As a matter of fact, with the exception of the “SpO2” channel, and the sleep stage information taken from the XML files, these were extracted exclusively from the “Pleth” signal. This strategy was chosen as it closely resembles the conditions in which this algorithm would later be tested in a real-world situation, as well as for the significant advantage of reducing the number of signals acquired and sensors used.

We utilise the EDF files’ HR signal to filter particularly noisy intervals by comparing it to the HR calculated using the PPG signal, as both signals are obtained through the same pulse oximeter, so if there is a considerable discrepancy between the two HR values, it could suggest that, regardless of whether it is due to the pre-processing, the noise contained in the signal, or the method of finding the peaks, the signal for this interval is unfit to be used. Similarly, the SpO2 signal was used to compensate for the lack of access to the secondary PPG necessary to calculate this variable.

While the data contained in the XML files did not require any kind of filtering or pre-processing, it is advantageous to manipulate the data in the EDF files before extracting information from them. The reasoning behind this pre-processing is to not only eliminate periods of signal that have too much noise to yield useful information or to improve the signal-to-noise ratio, but also to try to regularise the data between both different individuals and also different sources or datasets.

Although the advantages of the former are clear, the point of this regularisation is that by making the signals used as similar as possible by standardising amplitudes, baseline offsets, and other signal characteristics that we do not directly use as features to train our models, we do not change the information that is measured and extracted, just make this extraction easier and more reliable. Additionally, the results of applying filters become more predictable, allowing for a greater similarity of the extracted features in analogous situations, such as the same stage being detected between different individuals, which further improves the ability of the created models to learn from the signals. The ensuing process was only applied on the “Pleth” channel, as the “HR” and “SpO2” channels are already products obtained through processing applied on this former channel.

Accordingly, the first step of pre-processing was the standardisation of the signal, achieved through the subtraction of its mean, then division by its standard deviation.

After that, we analysed the signal in windows of 128 samples. To a certain point, this window size should be as low as possible and, after testing different window sizes, we found that using half the signals sampling frequency as window size worked optimally in this case. Afterwards, we calculated the mean value in each window, and once again removed it, in an effort to reduce the signal's baseline drift.

Subsequently, a 4^{th} order Chebyshev II bandpass filter, used as it is considered an optimal filter for short photoplethysmogram signals [77], when taking into account certain signal quality indexes [78], with a sampling frequency of 256 Hz (equivalent to the signal's own) and cut-off frequencies of 0.05 and 30 Hz was applied to the signal, with these values being chosen based on Kim et al. (2019) [79] and Pilt et al. (2013) [80].

Thereafter, in case we have access to accelerometer data (for example, when utilising the data that we acquired), we removed the intervals in which significant movement was detected, that is, when the signal's magnitude is above a certain, pre-defined threshold. At this point, the signal is deemed ready to have its features extracted.

3.3. Feature Extraction

For this thesis, we decided to work with features instead of just inputting the signal directly into the algorithm. The intent behind this is that, despite leading to an increase in the time spent during pre-processing, this procedure allows a greater degree of interpretability of the data utilised, as instead of having to look at several thousand samples if we desire to try understanding the reasoning behind a classification, we only need to look at a few dozen characteristics, which serves to reduce the "black box" nature of the algorithms. Additionally, since we will exclusively require these features to classify the signal, in case we are storing this data elsewhere, like in a storage server, for example, not only will these features take up less storage space, the amount of power and bandwidth necessary to transmit this information is lower than it would otherwise be. This will also serve to reduce the complexity of the models created later in this work.

After properly preparing the data, we proceeded to extract the relevant metrics from the signal. We start by determining the maximum, average, and minimum values of SpO2 and HR from the PSG files using their respective channels. Once done, within the defined 30-second intervals, we located the peaks in the previously processed PPG data. As a result of the pre-processing, the interval selection and peak detection were significantly

simplified, yielding very acceptable results which, in turn, increased the overall reliability of this process.

Having determined the number of peaks, we could calculate the maximum, mean [81] and minimum heart rate, and its standard deviation [82] using the PPG channel, at which point it can be compared with the HR information available in the dataset for a signal quality check, as previously mentioned. Subsequently, we used the sampling frequency and the position of the detected peaks, to determine features related to heart rate variability.

As stated earlier, features related to HRV were used as there are changes to the autonomic nervous system during sleep, with blood pressure, respiratory and heart rate adapting to the metabolic needs during sleep. As a consequence, the mean heart rate drops as sleep transitions from wakefulness to light sleep, and then to deep sleep. On the other hand, HRV increases significantly during REM sleep. These and other stage-specific variations make it possible to distinguish between different sleep phases [83, 84].

The resulting analysis of HRV is grouped under time-domain and frequency-domain [85].

The time-domain features extracted from the signal are based on the beat-to-beat intervals, that is, the time difference between a peak in the PPG signal and its preceding peak. To simplify the explanation of this procedure, and understanding the similarities in the information we are using from PPG when compared to ECG, we borrowed some terminology and reasoning from what is done for ECG. Knowing that RR intervals represent the time between each heartbeat, and are measured from peak to peak on the QRS complex that can be observed in ECGs (which is analogous to measuring the distance between peaks in the PPG signal), we add some filtering to remove artefacts and noise that would otherwise contribute to making these intervals unreliable. We did this by excluding RR intervals that are shorter than 0.15 seconds or longer than 3 seconds, or intervals where there is a RR interval with a larger difference than 150 milliseconds from 5 adjacent intervals. These NN intervals were used to calculate the following time-domain features:

- Root mean square of successive differences (*RMSSD*) [86];
- Standard deviation of successive differences (*SDSD*) [85];
- Number of pairs of successive NNs that differ by more than 50 ms and 20 ms (*NN50* [87] and *NN20* [86]);
- Total proportion of *NN50* and *NN20* in relation to the total number of NNs (*pNN50* [81] and *pNN20* [86]);

- Standard deviation of NN intervals (*SDNN*), in this case, calculated over the 30 seconds interval [86];
- Mean and median of NN-intervals (*Mean_nni* and *Median_nni*) [85];
- Coefficient of variation, equal to *SDNN* divided by *Mean_nni* (*cvnni*);
- Coefficient of variation of successive differences, equal to *RMSSD* divided by *Mean_nni* (*cvsd*);
- Difference between the longest and shortest NN interval (*range_nni*).

Contrastingly, frequency-domain methods assign bands of frequency and then count the total number of NN intervals that match each band. The frequency-domain features utilised were:

- Total power spectral density (*total_power*) [85];
- Power (or variance) in the very low-frequency band (0.003 to 0.04 Hz), which reflects an intrinsic rhythm produced by the heart, primarily modulated by sympathetic activity (*vlf*) [81];
- Power in the low-frequency band (0.04 to 0.15 Hz), that reflects a mix of both sympathetic and parasympathetic activity (*lf*) [81];
- Power in the high-frequency band (0.15 to 0.4 Hz), which reflects fast changes in beat-to-beat variability due to parasympathetic activity (*hf*) [81];
- Normalised *lf* and *hf* power (*lfnu* and *hfnu*) in addition to the ratio between *lf* and *hf* (*lf_hf_ratio*) [81], used by some investigators as a quantitative mirror of the sympatho/vagal balance [88].

Besides these features, information about the signal's entropy (specifically fuzzy entropy [89], dispersion entropy [90], approximate entropy [91], and sample entropy [92]) was calculated for each interval. As this information is time-consuming to obtain, we separately tested the amount of information gained about the sleep stages for each different kind of entropy calculated on a smaller subset of 200 PSG files, and chose to utilise only the features that provided a significant increase in the quality of prediction of the created models. After comparing the results, we opted for the use of both fuzzy and dispersion entropy. With the goal of further reducing the amount of time spent computing these features, without reducing their quality significantly, the average value of 32 sample long intervals ($1/8^{th}$ of a second) was determined, with this information being used to calculate these entropies instead. The choice of the window size used was obtained through trial and error and attempting to minimise time spent and information loss, in this

case, the average variation in the value of the calculated entropy features was around 10%, with a remarkable decrease in time spent (around 30 times faster).

Finally, as sleep is a continuous process, taking into account information about previous intervals is important, and, as such, for each interval we chose to save the two preceding stages' classification as well as the difference between these two values alongside the other above introduced features.

Throughout the feature extraction process, the quality of the acquired features in relation to the prediction of sleep stages was assessed by both the use of the software *Orange* and the creation and evaluation of models using the *scikit-learn* library.

In *Orange*, use of the *Rank* function allows us to rank each feature according to the amount of information that it carries or how important it is for the accuracy of the created models. We utilised this to avoid the addition of extracted features that do not contribute in a significant way to the prediction capabilities of the created models.

3.4. Methods of Classification

When developing models used to classify sleep stages, several different learning methods are commonly used. As a result of this, instead of focusing exclusively on one type of classifier, we opted to test most of the classifiers available in the libraries that we utilised.

3.4.1. Non-Neural Network Models

In regard to the non-NN models developed we evaluated the effectiveness of *Random Forest, Gradient Boosting, Gaussian Naive-Bayes, K-Nearest Neighbours*, and *Support-Vector Machine* classifiers.

During this thesis, we indirectly used *Decision Tree* learning as the basis of some of the ensemble methods, that is, methods that combine several base models to produce one optimal predictive model. This algorithm is a parametric, supervised (meaning that the datasets used to train the algorithms are labeled) learning approach. As during this work the target variable or label that we use only takes discrete values (correspondent to the different sleep stages), these *tree* models are called classification trees, where the *leaves* represent class labels, the *branches* represent conjunctions of features that lead to those class labels and the predicted outcome is the class to which the data that was being classified belongs. In essence, a feature (usually the one that can separate the most data) is chosen for each node, with this node branching into different resulting classifications (or

probabilities of classifications) depending on the range of values of the chosen feature. This branching process continues until a certain depth or efficacy is reached. They are one of the most popular algorithms due to their simplicity and intelligibility [93].

Random Forest is an ensemble learning method that constructs numerous decision trees at training time that can later be used for classification. Each of these trees is trained with a random set of variables, from which the feature that produces the most separation between its nodes is chosen. This serves to increase the diversity of results among the different trees, which together with bootstrap aggregation, allows us to further lower correlation across trees leading to increased performance when compared to single decision trees, as the ensemble prediction becomes more accurate than any of its individual predictions [94].

Similarly, *Gradient Boosting* is an ensemble learning method of weak prediction models, usually decision trees. Like other boosting methods, it involves incrementally building an ensemble by attributing a higher weight to misclassified instances during training, thus emphasizing model training for them. Essentially, instead of parallelizing the tree-building process in a similar manner to what happens during bootstrap aggregation, boosting takes a sequential approach where, for example, each decision tree attempts to solve for the net error of the previous decision tree, thereby reducing it. This means that, with careful tuning of its parameters, it may result in better performance than *Random Forest* models, but it can also more easily result in overfitting [95].

Conversely, *Gaussian Naive Bayes* classifiers are based on applying Bayes' theorem with a strong independence assumption to classify the data. The assumption that the selected features do not interact is unlikely in real data, but the approach still tends to perform well in most cases regardless. When dealing with continuous data, it is typically assumed that the continuous values associated with each class are distributed according to a Gaussian distribution, hence the name [96].

K-Nearest Neighbours (KNN) algorithms are non-parametric, supervised learning classifiers that utilise proximity to make predictions. For classification problems, a class label is assigned to a data element based on the vote of the K number of its nearest neighbours. Additionally, it is possible to construct a weighted version of this type of model, where instead of just taking the majority vote, the distance between the data points is taken into account, thus making the contribution of closer points more significant [97].

Finally, *Support-Vector Machine* (SVM) algorithms attribute classifications by finding a hyperplane in an N-dimensional space (with N being the number of features) that is able to

separately classify the data points. For a multiclass problem, the most common approach is the division of it into multiple binary classification problems [98].

As mentioned previously, ML algorithms have hyperparameters associated with them, and it is possible to improve a model's performance through the adjustment of these parameters [99]. To accomplish the fine-tuning of these models, we chose to utilise the *scikit-learn* library's implementation of the grid search technique (*GridSearchCV*), which uses cross validation. This, together with the *Orange*'s *Test and Score* and *Confusion Matrix* functions allow us to quickly gain an understanding of (beyond how relevant and adequate the metrics that were chosen to be extracted from the signal were, and how well pre-processed the signal was) what kind of model structure or method of model creation is optimal, as well as what kind of misclassifications are more common.

Similarly, the *scikit-learn* library allows us to easily create several different kinds of models and obtain their confusion matrixes, alongside a quantification of any other relevant desired metric, such as accuracy or F1-score, while, at the same time, being more customizable in terms of how each model is created when compared to *Orange* and providing an easy way to save such models to use and manipulate posteriorly.

3.4.2. Artificial Neural Networks

For this work, we also made use of artificial NNs, specifically *Multilayer Perceptrons*. The reason for using them is that, while each model has its advantages and disadvantages, NNs have the capacity to interconnect and map the input and outputs in a non-linear and much more complex way than other models (Figure 3.2).

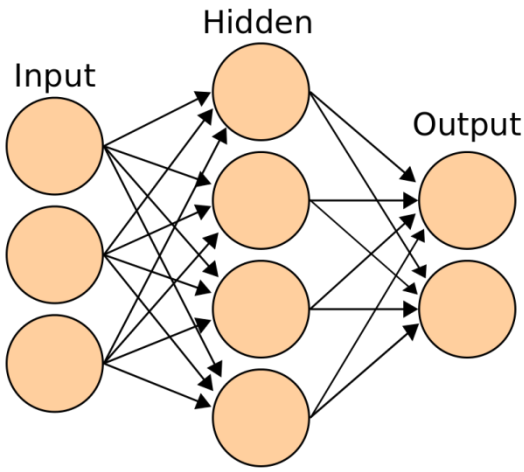

Figure 3.2- Illustration of the topology of a generic artificial NN with one hidden layer. Retrieved from [100]

Multilayer Perceptrons are a fully connected class of feedforward artificial NNs, consisting of at least three layers of nodes (input and output layers, with additional hidden layers in between). With the exception of the nodes in the input layer, each node is a neuron that uses a nonlinear activation function (such as rectified linear units or a sigmoid function). Learning in perceptrons occurs by changing the connection weights during training by comparing the errors in the output to the expected result [101]. This is done in *Multilayer Perceptrons* through backpropagation, with the rate or way the weights of the NN change being dependent on the chosen optimizer (with stochastic gradient descent or *Adam* being common choices [102]).

3.5. Minimization of Overfitting and Evaluation of Performance

Throughout the development of the algorithms in this work, we took care to calculate the models' accuracy values based on their effectiveness on files that they had not previously been exposed to, so as to get a more accurate portrayal of the real-world accuracy of the models by minimising the possible effects of overfitting. In a similar manner, while simultaneously taking into account and attempting to maximise the models' performance (which, to a certain point, should increase alongside their complexity), we also considered it important to try reducing their complexity as much as possible, since, as formerly explained, this is central in reducing overfitting.

For the NN, to further these goals, several attempts at attaining this reduction were done during this work, for example, when we chose to, whenever appropriate, minimise the number of hidden layers and neurons used to create them, or when we tested the results of applying regularisation techniques on the developed structures.

One of the studied regularisation techniques applied was dropout. Dropout is considered a noise injection technique, and works by randomly (with a certain, pre-determined probability) ignoring some number of layer outputs during training [103]. The reasoning behind this regularisation method is to alter the layer connectivity and, as such, search for alternative paths to convey information from one layer to the next. This leads to each update in a layer during training being performed with a different visualization of the layer itself, which aims to approach the Bayesian gold standard of regularising a fixed-size model (that is, training a large number of NNs with different architectures in parallel, averaging the predictions of all possible settings of the parameters and then weighting each setting by its posterior probability given by the training data), while using significantly less computation [104].

As such, besides minimizing overfitting, dropout also serves to approximately combine several different NN architectures efficiently, and given that model combination consistently improves the performance of ML methods, but is avoided due to being cost-prohibitive, this is a technique that possibly addresses both these issues successfully [104].

Other regularisation techniques that control model complexity by adding an extra penalization term at the end of the loss function, such as L1 (lasso regression) [105] and L2 (ridge regression) [106] regularisation, were also evaluated.

The main difference between L1 and L2 techniques is the penalty term, where the first adds an absolute value of the magnitude of the neuron's weight as a penalty term to the loss function, while the latter adds a squared magnitude of this weight as a coefficient. This means that, while L2 works very well to avoid overfitting since it has a more pronounced effect for high network weights, L1 shrinks the weight of the least important features to zero, essentially removing them, meaning it is somewhat more appropriate for feature selection.

Both of these techniques can be modified by lambda or rate, which adds more or less weight to the cost of this regularisation element in the loss function. As a high rate may lead to too much weight being attributed to this element, which can cause underfitting, the selection of a proper lambda is important, especially in the case of L2 regularisation [105].

Since L2 regularisation appends the squared value of weights in the cost function, it has a substantially more pronounced effect on the directions of the weight vector that contribute less to the loss function, when compared to its effect on other directions that do. This as a result reduces the variance of the model, without increasing its bias significantly, making it easier for the model to generalize on unseen data.

The choice of the loss function is also crucial for the development of the algorithms [107], as it is a method of evaluating how well an algorithm does in terms of predicting the expected outcome of a dataset [108]. During training, the goal is the minimization of the error between the actual and predicted outcome, which in practice means we desire to minimize the value of the chosen loss function.

Beyond the selection of these parameters, suitably determining and comparing the actual level of performance of the created models is also important. One issue is that, usually, models are stochastically trained, meaning that two models with the matching architecture being trained with identical data in the same manner, might perform differently after training, which may further complicate the study and understanding of the training process. One way to deal with this issue is simply constructing several identical

models with the same characteristics, evaluating the average of their performance, and then using these results to compare with the average performance of other models with different architectures.

Another difficulty is that, despite the high interpretability of accuracy, it is not a very appropriate metric for measuring the performance of algorithms in unbalanced datasets, which, as mentioned previously, is the case for normal sleep. This is problematic as models' learning on unbalanced data may start neglecting to learn about the least represented classes, as, besides just having less information that the models can use to train with, having a poor classification performance on these classes does not penalize the models' accuracy very significantly (even if its performance is worse on a class, if it is less represented the total number of misclassifications might be lower still). This means that the models' calculated accuracy loses its relevance as a proper evaluation metric for our purposes, since, for instance, despite consistently failing the classification of 3 out of its 5 classes it may nevertheless present a high accuracy.

For this purpose, F1-score might prove more useful, as, while we balanced the datasets used for training and testing, in a real-world scenario the various sleep phases are unbalanced, and so, since F1-score is more sensitive to data distribution and more heavily penalises false-negatives it might be preferable to accuracy. Similarly, since during our work we will be comparing the results we have obtained to ones that were classified by technicians, determining the measure of agreement between both might be useful. For this end, we propose the use of Cohen's kappa coefficient. This coefficient determines whether the degree of agreement between two raters is higher than would be expected by chance [109], being one of the most important and widely accepted measures of inter-rater reliability [110].

3.6. Corroboration of Results through Collection of Real-world Data

After achieving some measure of success in the creation of the algorithms for sleep stage prediction, we proceeded to acquire our own data, classifying the sleep stages and then comparing them with the classification attributed by a validated sleep prediction algorithm available in the market, with the intent of determining, in a less controlled environment, the comparative effectiveness of our algorithm. For this end, we used a device developed by PLUX, biosignalsPlux, which allows for high-quality biosignal acquisition of 8 channels in up to 3000 Hz sampling rate while having 16-bit resolution per channel [111], and a TicWatch E2 smartwatch.

PLUX was established in 2007 and creates innovative products for healthcare and research by developing advanced biosignal monitoring platforms that combine wearable body sensors, wireless connectivity, and software applications to deliver valuable products for their target markets. They aim to create miniaturized sensors and wearable devices for biosignal acquisition and processing with maximum comfort and safety, with user-friendly software [112].

The biosignalsPlux device was used with an accelerometer and blood oxygen saturation attachment, sampled the signals at 200 Hz, and supplied the resulting data to the developed models for sleep stage prediction. One of the advantages of using this device compared to many other wearables is that it gives us access to the biosignal's raw data, allowing us to more conveniently adapt it to our algorithm's needs. This is also an important standard for the field as a whole, as it allows algorithms to use raw data for training and classification, instead of being limited to using the data that other wearables provide. As mentioned previously, this limitation is an issue, because this information is often uniquely processed (due to the specific hardware and software utilized by these devices), thus making the validation process more difficult.

Contrastingly, TicWatch E2 was used in conjunction with a mobile phone application called "Sleep as Android" [113, 114]; using its accelerometer and PPG data to assist in the sleep stage classification done by this app.

The smartwatch was used on the wrist of the dominant hand with the PPG sensor placed on its posterior side. On the other hand, the biosignalsPlux's PPG sensor mirrored its positioning, the accelerometer sensor was placed on top of the PPG sensor, and both sensors were fastened by a wrist brace.

As the sources of this data are different from the dataset used to train the algorithm, despite most of the pre-processing being done in such a way to be easily generalizable, some slight adjustments to the pre-processing were still necessary to be able to better analyse the data acquired this way. More specifically, as mentioned previously, in this case, as opposed to the MESA database, we have access to accelerometer data, which was collected and used to both more easily identify the start and end of the sleeping interval, as well as facilitate the removal of noisy intervals of sleep. In the case that intervals in the middle of sleep were removed because of this, in the effort to keep both records synchronized, they were removed in both of the recordings. In addition to this, we only utilised records that, after being successfully synchronized, had a duration of at least five hours. In total only 14 nights (approximately 80 hours) from one individual matched this criterion.

The data obtained through the biosignalsPlux device was acquired through "OpenSignals (r)evolution", a Python-powered software that can be downloaded for free from the biosignalsPlux website [115]. This data is then saved in a TXT file and a HD5 file, with the relevant signal data being easily accessible in the TXT file, ready for pre-processing and, afterwards, classification by the chosen algorithm.

Subsequently, after pairing the smartwatch with a phone, the data captured by it was imported through the use of "Google Fit" [116], with all resulting sleep stage data being stored in a single JSON file. This information was read, split into the different nights, and stored in CSV files to facilitate its visualization through graphs (such as the one in Figure 3.3b), and comparison to the sleep stages classified by our algorithm.

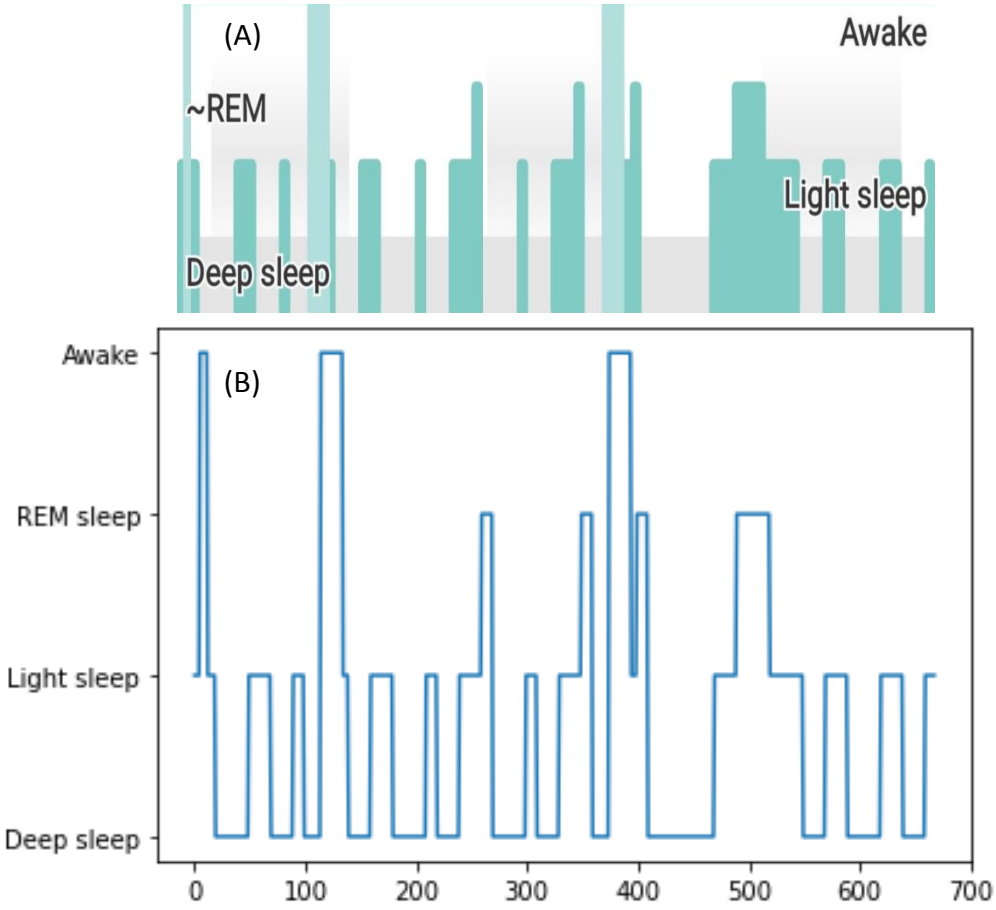

Figure 3.3 – Example of a graph produced by a night's sleep through the phone app "Sleep as Android", A), and the same data visualised through Python, B).

As can be seen in Figure 3.3, "Sleep as Android" only classifies sleep into 4 different stages, with only one stage corresponding to light sleep. As our algorithm classifies into 5 different stages, to simplify comparison, stage 1 and stage 2 were converted into a single stage of light sleep.

4. Results and Discussion

4.1. Non-Neural Network Models

As mentioned previously, for the creation of the non-NN models we used *scikit-learn* [117]. The data utilised to train these models was split utilising its *train_test_split* function. This split was stratified so that both the training and test sets had the same proportion of sleep stages, and, initially, 80% of the data was used to train the algorithms, while the remaining 20% of data was utilised as testing data. Besides this split, we also separated some files from the training set and used them exclusively for a test set, which was later used to validate the created models more rigorously.

To avoid the occurrence of imbalanced learning we chose to balance the data. This was done through different means, depending on whether the data belonged to the testing or training set. The data we used to train the models with was balanced via oversampling, i.e. we duplicated data pertaining to intervals that belonged to the least common sleep stages, while data in the test set was balanced by undersampling, that is, intervals belonging to the most common sleep stages were discarded until every class was equally represented. Usually, as we have a great amount of data available to train the models with, we could use undersampling for both, however, as after a few tests no increase in overfitting was noticed as being caused by this choice, and, to avoid unnecessarily discarding information, we decided to oversample the training files instead, while still undersampling the files used for testing, as we should not care as much about minimizing loss of information in this case.

To speed up the process, these initial models were trained only using data belonging to 200 files (equivalent to around 1700 hours of sleep). For this same reason, we did not begin a more in-depth analysis of the model creation process until the ones developed by resorting to the default *scikit-learn* settings achieved around 80% accuracy on the training set. This threshold was chosen as only at this point did we consider to have gathered enough evidence that the models were able to sufficiently learn from the acquired features. Instead, until we attained this target, we chose to proceed with the addition of new features and adjustments to the pre-processing.

Of all the trained models, the best performing models were the ones using the *Random Forest* classifier, with one such example displaying 94.53% accuracy at the end of training by *scikit-learn* (for the sake of reference, the *Gradient Boosting* attained 89.91% accuracy, the *Gaussian Naive-Bayes* reached 61.73% accuracy, the *K-nearest Neighbours* displayed 33.00% accuracy, and the *Support-Vector Machine* achieved 35.34% accuracy). Upon

testing this *Random Forest* model on the test set, containing another 200 files, to which the training set did not have access, this precision dropped to 71.39% with a Cohen's kappa of 0.64, as it is possible to observe in the following graphs (Sub-Figures 4.1a and 4.1b).

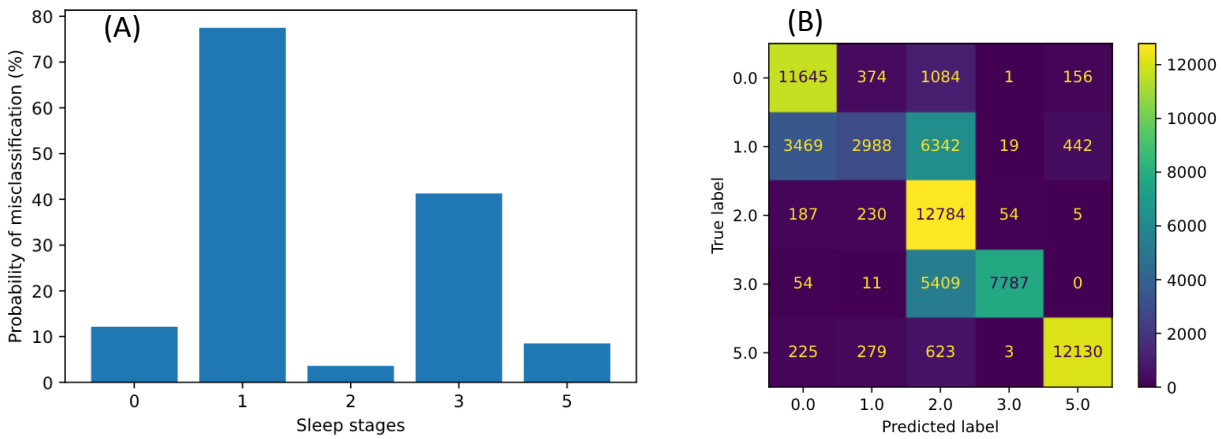

Figure 4.1 – (A) Graphical representation of the probability of misclassification for each sleep stage. (B) Confusion matrix of the created model for the test set.

As we can see from these figures, the performance of the model is significantly lower for sleep stages 1 and 3 when compared to stages 0, 2, and 5. A possible explanation for this is that the natural disparity in the number of intervals with stages 1 or 3, when compared to stages 0 and 2, is large enough that even after balancing, the model is unable to properly learn how to classify these stages. Meanwhile, stage 5 is so distinct from the other sleep stages, that even with a reduced amount of information the model is still able to learn how to differentiate it from other stages.

With this in mind, in an attempt to reduce the model's overfitting, we reduced the amount of data utilised during training, with 90% of the data used as the test set, and only 10% used to train the model. Once again, the best performing model was *Random Forest*, displaying this time an accuracy of 83.2% on the training set. Running this model on the 200 separate test files showed an increase in performance to 76.40% overall precision, 5.00% higher than when compared to the previous model, and a Cohen's kappa of 0.70 (Sub-Figures 4.2a and 4.2b).

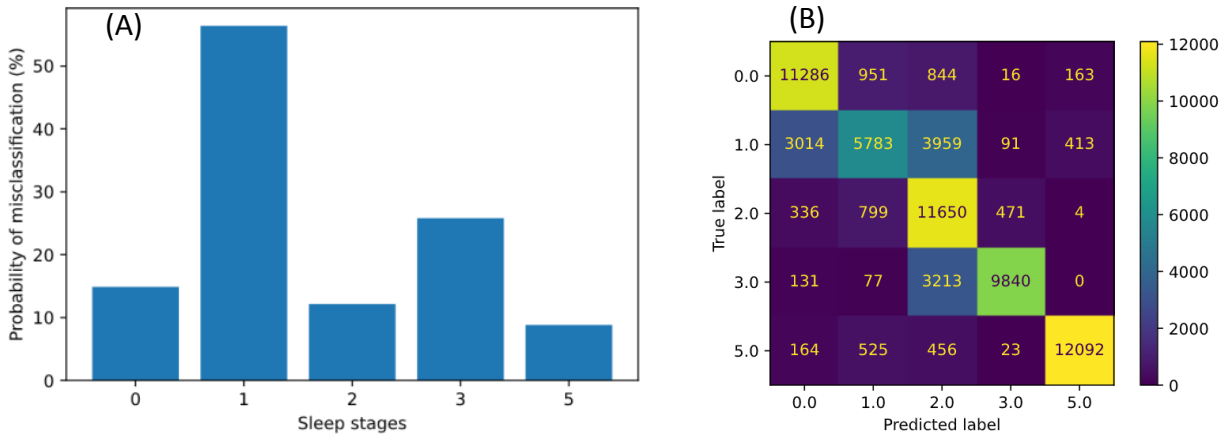

Figure 4.2 – (A) Graphical representation of the probability of misclassification for each sleep stage. (B) Confusion matrix of the created model for the test set.

Through observation of these graphs, we notice that the model's performance on classifying stages 1 and 3 intervals increased significantly (from 22.53% accuracy to 43.61% for stage 1, and from 58.72% to 74.20% for stage 3), while its performance for classifying stage 2 and stage 0 lowered (from 96.41% to 87.85% for stage 2, and from 87.82% to 85.11% for stage 0). Stage 5 classification remained mostly the same, dropping only very slightly in performance from 91.48% accuracy to 91.19%.

We believe that this overall increase in performance is due to the fact that the intrinsic interpersonal variance is significantly higher than the average variance in the different intervals belonging to the same stages of the same individual. Because of this, while in theory, a reduction in the quantity of information used to train the algorithms would lead to a decreased quality of prediction, in reality, using too much information from an individual leads to slightly lower precision when compared to reducing the amount of data utilised from each individual, but instead increasing the amount of data used by sampling more people.

This means that despite there being an apparent drop in the model's performance when being tested on files that were used for training, it can better distinguish the different sleep stages on files from individuals it has not seen before, which is what is desired from the development of these models.

This is a slightly different situation than normal overfitting, where, because a model trains for too long on sample data or becomes too complex, it starts learning from the noise that exists in the data or otherwise irrelevant information. In this case, we think that this effect can be mostly explained by the fact that each PSG file originates about two thousand data points, many of them duplicated because of the discrepancy in frequency between the different stages, the models might end up attributing undue importance to

characteristics innate to the individual that would not otherwise matter for classification as they are not shared between individuals, and if this is the case, simply increasing amount of data made available to the model would not help mitigate this issue, as we would have to ensure that new data comes from different individuals than the ones already sampled.

As such, reducing the percentage of data points utilised for training presents itself as a suitable solution for minimizing this kind of overfitting, but still does not remove the necessity of separately testing the models with files from different individuals than the ones they were trained with, as the displayed accuracy at the end of training by these libraries can otherwise be misleading. We also predict that this effect would be significantly reduced by using a larger set of data, possibly even justifying a more even split of the training and testing sets.

That being said, it is important to point out that the model's performance on an entire PSG actually decreased on the second type of model despite the overall performance seemingly increasing. This is the direct consequence of the unbalanced distribution of sleep stages, with stages 0 and 2 being substantially more common than stages 1, 3, and 5.

Thus, despite there being a more significant increase in model performance when it comes to stage 1 and 3 detection, than a reduction in performance when predicting stages 0 and 2, once utilised in a real-world situation the amount of misclassifications is considerably higher.

We can observe this in Sub-Figures 4.3a and 4.3b, which were produced through the creation and comparison of maps of a test file's sleep stages over time, where each point represents a 30 seconds interval.

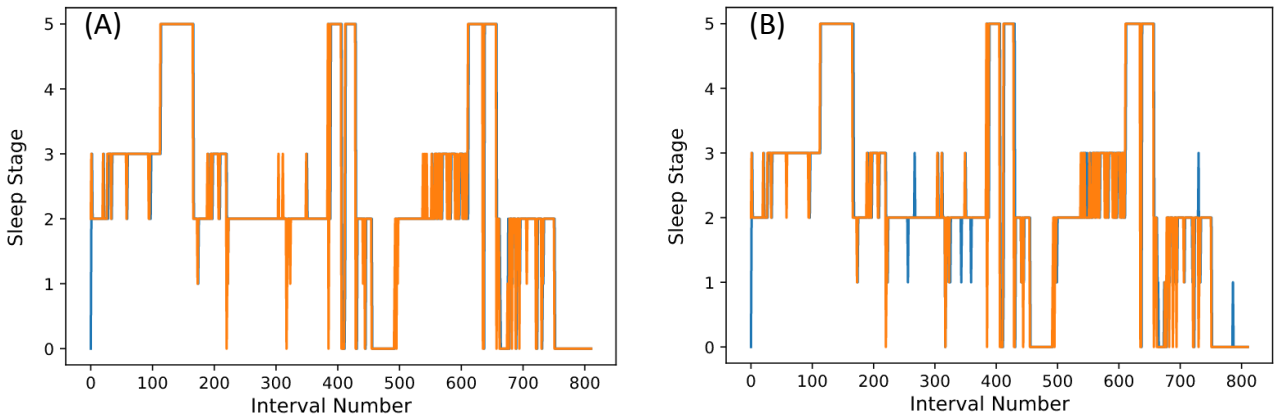

Figure 4.3 - (A) A randomly chosen file, processed by a model created with a 90% training, 10% testing split; and (B) The same file, processed by a model trained with a 10% training, 90% testing split. The orange lines represent the actual stage attributed by the technicians to the intervals, while the blue lines are the stages predicted by the model. If the blue lines are not visible, that means the maps are perfectly aligned, that is, the prediction was correct.

Because of this, while it might seem that the second model's performance is worse as it fails more often in the most common classes, because every stage has its physiological importance, and to try to preserve the sleep structure as close to reality as possible, we consider improving worst-case classification performance as important, even if the model's performance for stage 0 and 2 prediction worsens slightly, which is where F1-score becomes an important metric.

With this information in mind, in an attempt to further improve model performance we used grid search and all of the recorded files in the training folder. The chosen criteria were selected based on the parameters *scikit-learn* allows us to change for each model (Table 4.1), with the results of the best performing models being displayed in Table 4.2.

Table 4.1- Values for the different parameters to be optimized when utilizing gridsearchCv.

	Parameters	Values
random forest	n_estimators	10-100 and 100-1000 (in increments of 10 and 100, respectively)
	criterion	gini, entropy
	max_depth	None, 10-100 (with 10 step)
gradient boosting	n_estimators	10-100 and 100-1000 (in increments of 10 and 100, respectively)
	criterion	friedman_mse, squared_error, mse
	max_depth	1,3,5, 10-100 (in increments of 10)
KNN	n_neighbours	1, 3, 5, 7, 9, 11
	weights	uniform, distance
	metric	manhattan, euclidean
SVM/SVC	C	0.0001, 0.01, 0.05, 0.1, 0.5, 1.0, 5, 10
	kernel	linear, poly, rbf
	gamma	scale, 0.0001, 0.01, 0.05, 0.1, 0.5, 1.0, 5, 10

Table 4.2- Values of the chosen metrics for the highest performance models of each type.

	Balanced dataset		Raw, unbalanced files		
	Accuracy (%)	Cohen's kappa	Cohen's kappa	Macro average F1-Score (%)	Lowest class accuracy (%)
Random Forest	79.30	0.7412	0.7255	75.97	42.11 (class 1)
Gradient Boosting	82.34	0.7792	0.6967	75.40	56.32 (class 1)
Gaussian Naive-Bayes	68.41	0.6052	0.5285	61.71	33.36 (class 1)
KNN	21.99	0.0249	0.0219	19.71	14.54 (class 5)
SVM/SVC	25.03	0.0628	0.0564	21.53	16.47 (class 1)

As can be observed in Table 4.2, increasing the number of files available for the training of these models and using the grid search method to find optimal values for several of the models' hyperparameters lead to an increase in the performance of most models. It is possible to notice that by a significant margin the best models are *Random Forest* and *Gradient Boosting*, with this last model presenting an overall more balanced performance for all of its classifications when compared to the other models and presenting the highest accuracy and Cohen's kappa for the balanced dataset, with the former model, on the other hand, having a better performance for the unbalanced dataset.

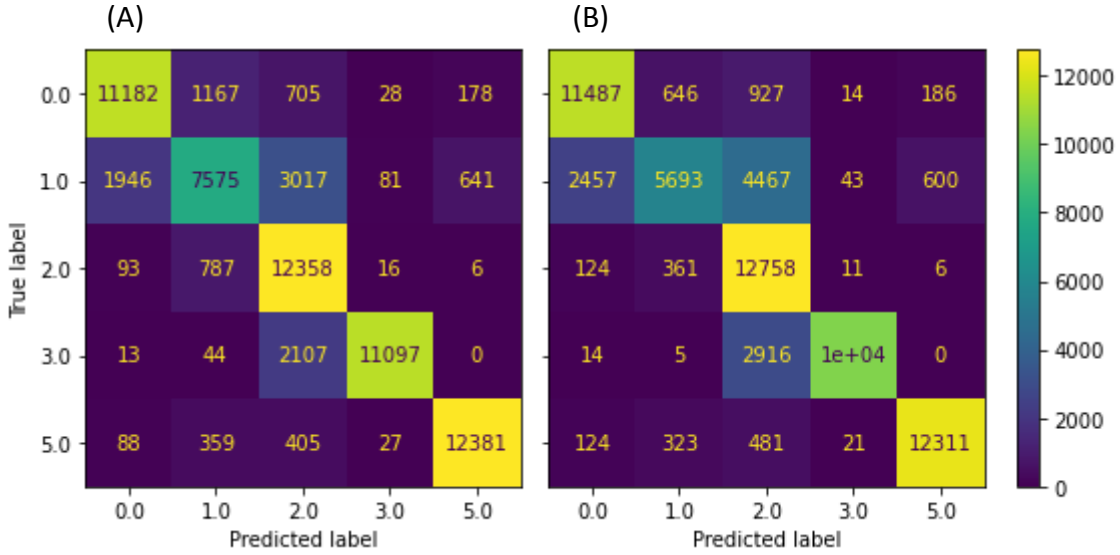

Figure 4.4 – Confusion matrix of the *Gradient Boosting* model, A), and the *Random Forest* model, B), for the balanced test set.

As the behaviour of both models is similar, to further distinguish them, we suggested the use of a specific cost matrix, with distinct costs being attributed to different misclassifications, depending on how similar the true and predicted class are or how "far apart" they are from each other (Table 4.3).

Table 4.3- Suggested cost matrix.

	Predicted label				
True label	0	1	2	3	5
0	----------	0.5	1	2	1
1	0.5	----------	0.5	1	2
2	1	0.5	----------	1	2
3	2	1	1	----------	2
5	1	2	2	2	----------

After applying the cost matrix, *Gradient Boosting* has a lower associated cost, meaning, in theory, that even if the rate of misclassification is similar, the overall importance of the mistakes is lower (since, for instance, we considered that the model mistaking stage 1 with

stage 2 (both part of light sleep) is less concerning than misclassifying stage 1 as stage 3 (which is deep sleep)). Consequently, we selected this model as the best performing non-NN model developed so far.

4.2. Creation of Deep Neural Networks

As shown throughout this work, the structure and way ML models are trained are key in reducing misclassifications. As such, in the effort to further fine-tune the model creation process, how it learns from data, and the suitability of the model to the problem, we decided to create NNs, which, as they allow for the creation of models with greater complexity, should enable us to attain greater effectiveness of the algorithms.

Assuming that the data utilised to train a NN has enough information to be used in the prediction of the data's label, in this case, the interval's sleep stage, the NN's performance is dependent on its structure, with the suitability of said structure being dependent on the problem being solved and information available to it [118].

To our knowledge, even considering other studies done in this area or with the same datasets (such as Sun et al. (2020) [119] and Sridhar et al. (2020) [120]), this exact problem with this set of features has not been solved before and, because of this, the optimal structure of our desired NN is unknown. As these problems are commonly approached by resorting to trial and error, in this work scripts were developed to run through a large number of different structures, with both different parameters and hyperparameters, briefly training each one and comparing how well they perform.

For the creation of these NNs, we used *Tensorflow*. *Tensorflow* was chosen as it is a low-level library that provides more flexibility and network control. This allows us to not only define our own functionality or services for the models, which facilitates their adaptation based on changing requirements, but also helps us more easily understand how operations are implemented across the network, and to directly visualize the created ML models with its built-in tools. Additionally, we can extract and save these models for later use, which would assist in a possible future integration of the algorithms in device development.

For this work, all created NNs were composed of *Dense* layers (meaning that all neurons from one layer are connected to all neurons from the previous and next layer) and had *relu* (rectified linear) activation functions for both its input and hidden layers. The input layer had a width equivalent to the total number of features, with the same number of neurons as the succeeding layer. As for the output layer, its activation function was instead *softmax*, which, when coupled with the fact that its neuron count matches the

number of labels, essentially means this last layer will output class probabilities. This is a requirement as we have chosen to utilise categorical cross-entropy as our loss function, which trains NNs to output a probability over the 5 classes for each 30-second sleep interval.

The optimizer chosen to train these models was *Adam* with a starting learning rate of 0.001. The chosen loss was categorical cross-entropy and the selected metric during compilation was accuracy on the validation set.

To start, we created 2048 NNs for each of four different structure types, depending on how the number of neurons varied between each layer. In homogeneous patterns, each layer would have the same number of neurons per layer, in diminishing ones, each layer would inherit 75% (rounded to the nearest integer) of the number of neurons of the previous layer, in increasing patterns each layer would have 25% more neurons than the previous layer, and finally a random pattern, where, in each layer, the total number of neurons would be randomised.

According to these different structures, a nested for-loop sequence was used, described in further detail through pseudocode in Algorithm 4.1.

Algorithm 4.1- Pseudocode describing the nested for-loop sequence used for inputting the starting neuron and layer numbers for the different model architectures to the model creator process.

```
for number_of_hidden_layers := 1 to 64 do:

    for initial_number_of_neurons_per_layer :=16 to 512 by 16 increments do:

        layers = number_of_hidden_layers
        neurons = initial_number_of_neurons_per_layer
        train_neural_network(layers, neurons, epochs = 30, file_count = 200)

    End
End
```

The increment of 16 per loop for the number of neurons was chosen since exhaustively exploring all these variables one by one would take too much time, and unnecessary since the purpose of this test was simply to narrow down the possible architectures into an acceptable baseline foundation. Each NN was trained for a total of 30 epochs, and, similarly to before, only the data from 200 files was utilised so as to reduce the time spent in this stage.

As a result of this trial, models created with a homogeneous structure proved to be generally superior, with four out of the five best models being of this structure type, while the only other model type having a result in the best five was a model with the randomised structure. Other patterns were also observed, such as the generalised drop in model performance when the total number of hidden layers surpassed ten, with most models above this number, regardless of structure type, presenting a similar accuracy to just random guessing, no matter how long they were trained. The best performing models with the homogeneous structure were two models with three hidden layers (one with 32 neurons and another with 128), and two others with seven hidden layers and the same neuron configuration as before.

In order to test if the larger, more complex models were underfitting to the data due to the number of epochs, five models with 32 layers and 128 neurons per layer were trained for 500 epochs, showing no improvement in performance over this period. Other tests to try resolving this underfitting were conducted by raising the starting learning rate of the *Adam* optimizer to 0.01, and, later, to 0.1, changing the *Adam* optimizer to a stochastic gradient descent (SGD) optimizer with 0.01 learning rate, and then with a learning rate of 0.1 and a momentum of 0.9. Despite these attempts to optimise the training of these larger models, they failed to learn from the data.

A possible explanation for this is that the created NN is too large and complex for the problem being solved with the selected features, such that the exceedingly small variation in the weights of the neurons caused by the learning process has too small of an impact on the overall performance of the network, effectively impeding the learning process (as these are NNs using gradient-based learning methods and backpropagation, this could be explained by the vanishing gradients problem [121], and if so, this could be mitigated through use of the batch normalization [122]).

Following this, one of the best performing models obtained from the previous trial (as the performance of these models was similar, the simplest and smallest one, with 32 neurons per layer and with 3 layers, was selected) was chosen as the standard model henceforward, with the next step intending to fine-tune the structure of the network, while still accounting for the inherent randomness of the learning process.

After limiting the number of neurons per layer to 32, we trained 22 different NNs with varying layer numbers over 200 epochs, repeating this for each architecture ten times in total. We tested these models in the previously chosen, separate, test set of 200 files, and saved the resulting values of maximum accuracy, minimum loss, and in which epoch these values were achieved, with the following results (Figure 4.5).

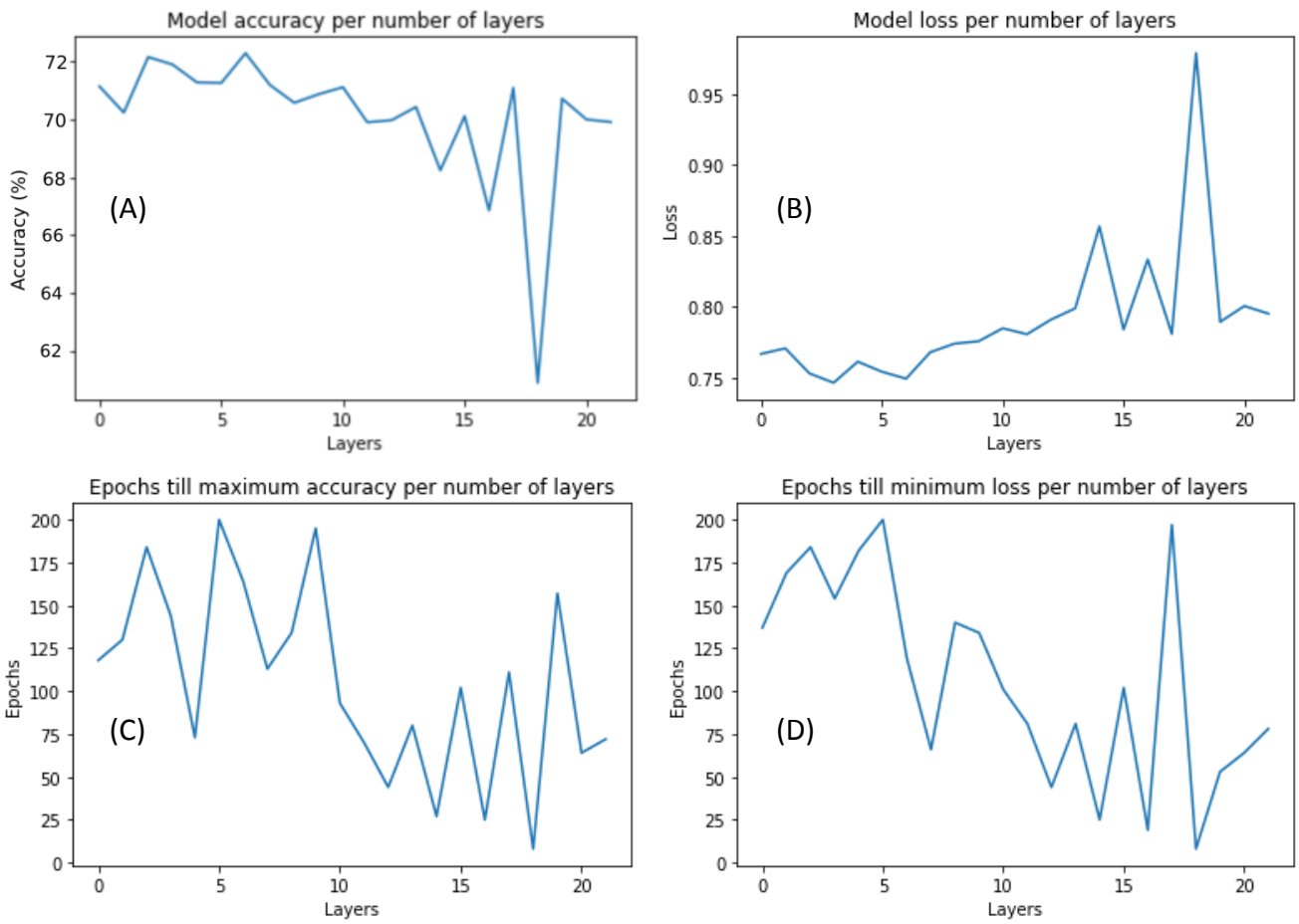

Figure 4.5 - Composite image of model accuracy, loss, epochs till maximum accuracy and minimum loss for models with 32 neurons per layer and increasing hidden layer number.

In these graphs is possible to see that the models' quality is highest for models with 3 and 7 layers, confirming what was observed in the earlier test, with models tending to worsen in quality once they cross this threshold. To note, we chose to train the models for a high number of epochs, disregarding any possible resulting overfitting, as, since we are recording the maximum accuracy and minimum loss over this period in the test sets, it should not be a concern. This would also allow us once again to test if the low performance of the more complex models is due to being limited to a shorter training time than they would otherwise need. However, as can be seen, these larger models tend to reach their best performance relatively early when compared to the other models, indicating a more limited capability to learn from the datasets, especially when their lower accuracy is also taken into account.

In order to better show this, we saved information about the learning process from two of the trained models, one with three layers and another with 22 (Figure 4.6).

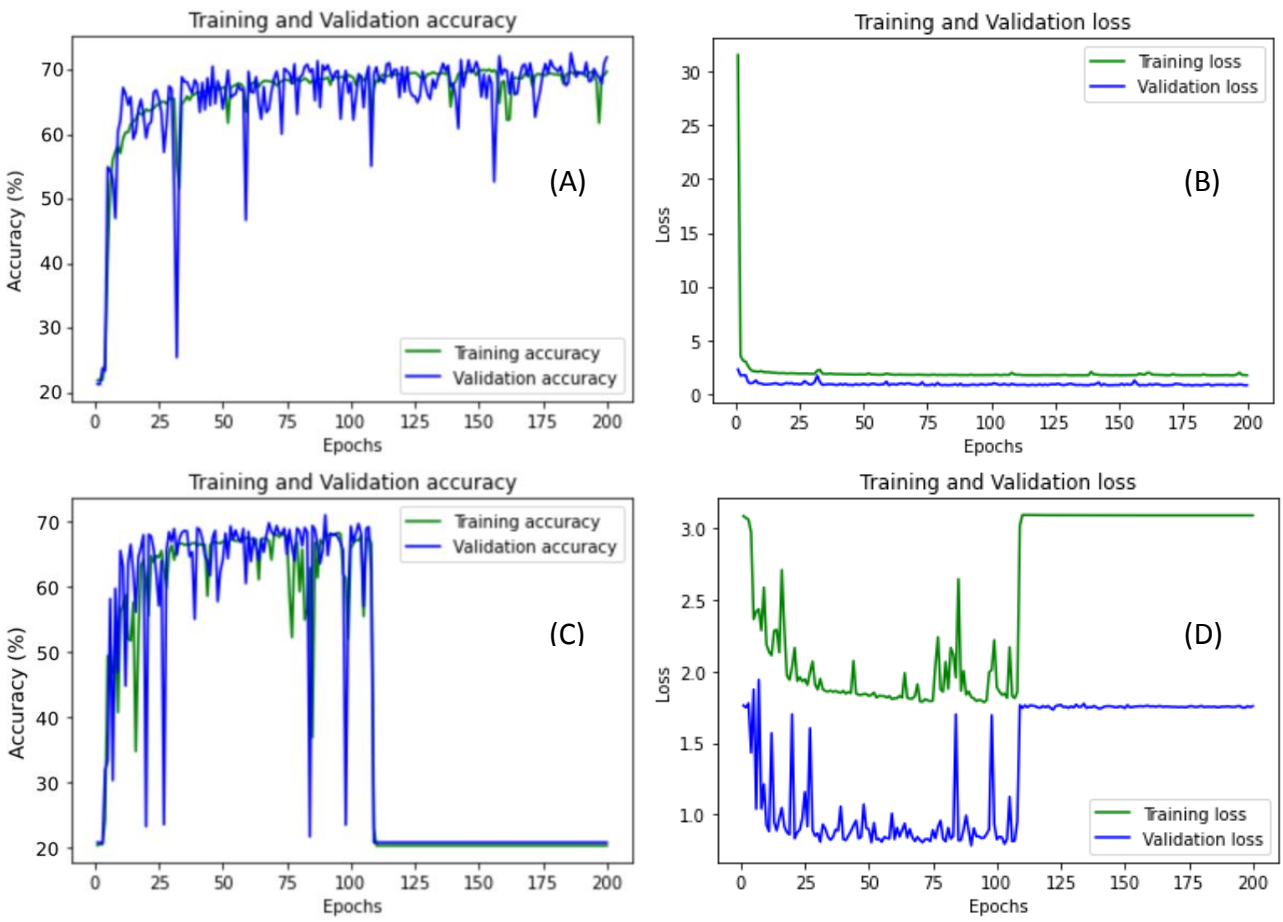

Figure 4.6 – Composite image of model accuracy and loss over 200 training epochs, for the models with 3, A), B); and 22, C), D) layers.

From these images it is possible to see that the performance of more complex models is less stable, with more frequent drops from around 70% accuracy to lower than 30% during training. Additionally, besides the maximum accuracy being lower, with the 22-layer model presenting a maximum of 69.90% accuracy and the 3-layer model having 72.10%, it is possible to observe that, at a certain point, the models' performance drop to the quality of a random guess permanently (this happened on epoch 107 where the precision first dropped from 67.25% to 48.19% on the following epoch and then to 20.34% at which point it stabilised). This sudden accuracy drop was never observed during the training of the 3-layer model, which, simultaneously, showed no evidence of overfitting over the 200 epochs.

With the completion of this test, we limited the number of layers to three, and performed a new optimization step to find the optimal number of neurons. To do this, we built another for-loop that created models with three layers and with a variable number of neurons in each layer, with the first model having two neurons per layer and the following models having twice as many neurons in each layer until the last one that had 1024 (Figure 4.7). This increment rate was chosen as it allows us to test over a wide range of values,

while concurrently having a lower resolution for higher numbers of neurons, which we know from previous trials do not perform very well, and a higher resolution for lower numbers, which privileges the creation of simpler models. This process was repeated ten times, with the best performing models being selected to be represented, in an attempt to minimize outlier effects caused by lack of determinism in model creation.

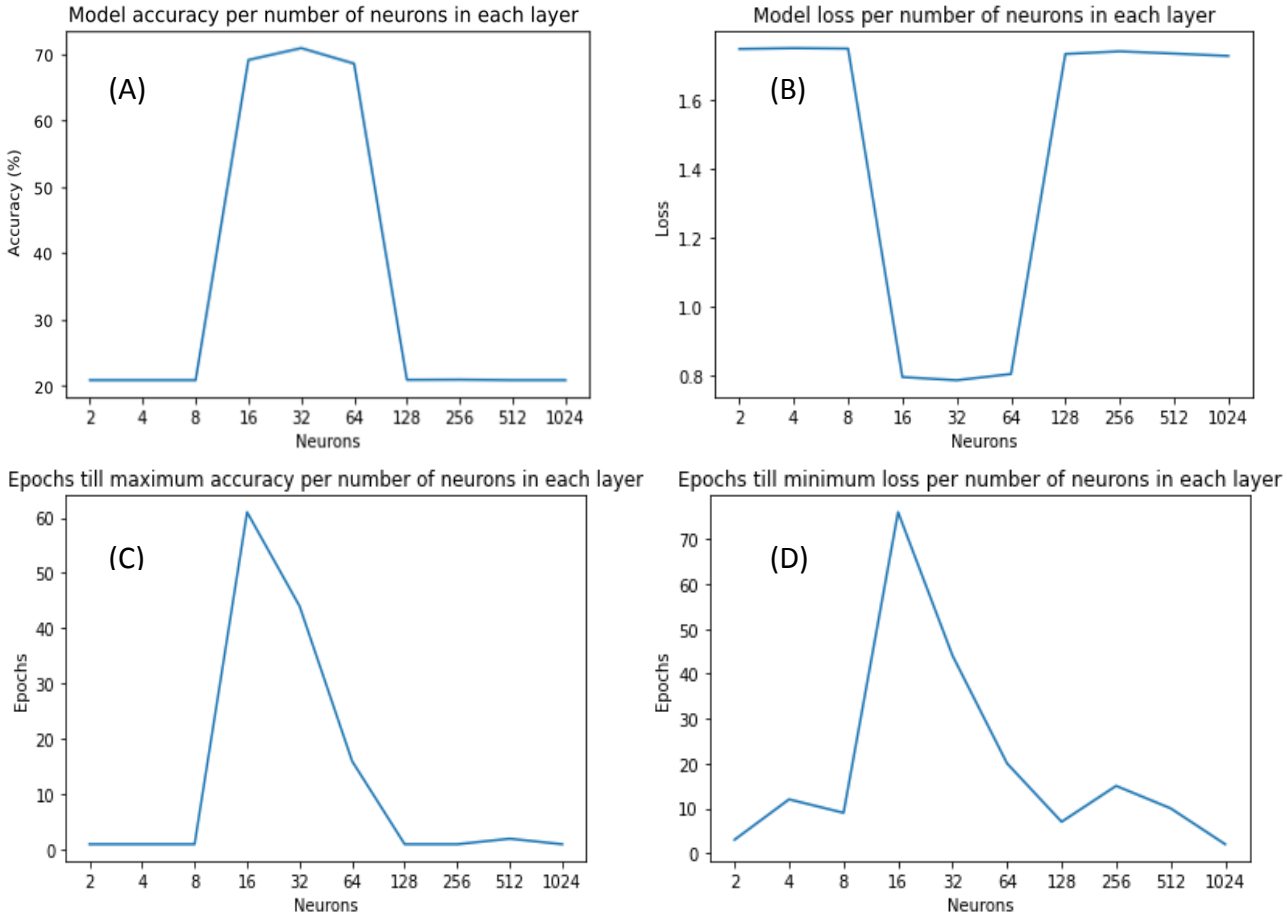

Figure 4.7 - Composite image of model accuracy, loss, epochs till maximum accuracy and minimum loss for models with 3 layers and an increasing number of neurons per layer.

As we can see from these graphs, models that have less than 8 or more than 128 neurons per layer appear to be unable to correctly learn from the data, while models that have between 16 and 64 neurons per layer, exhibit the best performance of all models.

Similarly to before, the least adequate models tend to reach their best performance comparatively early, indicating a limited ability to learn from the data, while also suggesting that this underfitting is not due to low training time.

To further fine-tune the previous test, we created more 3-layered models, this time with neurons per layer ranging from 16 to 64 and incrementing this number by one for each model. Similarly to before, we repeated this process ten times, selecting the best results for each different model configuration (Figure 4.8).

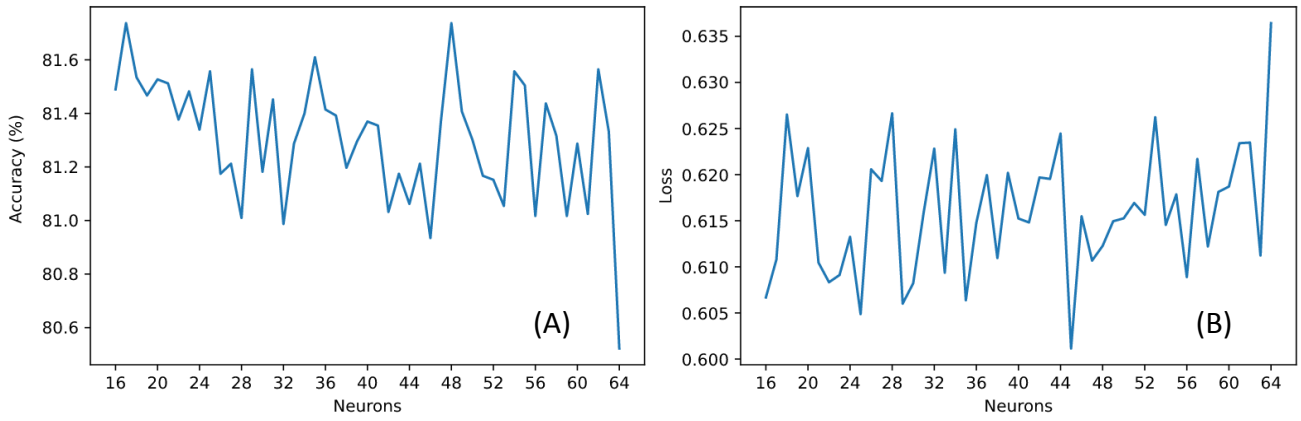

Figure 4.8 – Highest accuracy, A), and lowest loss, B), of the created models over the 10 separate training attempts, according to the number of neurons per layer.

As the performance of these models is similar (with the accuracy fluctuations between most models being low enough that it can be attributed to stochastic differences in training), this criteria is no longer useful for determining the optimal NN structure. At this point we could merely select the simplest of these models (so 16 neurons per layer) and use it as the structure for the final algorithm; however, during training we noticed that some of these models were distinctly unstable performance-wise, with some of these structures only having an acceptable level of performance (>70% accuracy and >0.60 Cohen's kappa) on very few of the created models. As this lowers the average performance of these models, for the purpose of optimisation of the training processes of similar models, we chose to further explore this (Figure 4.9).

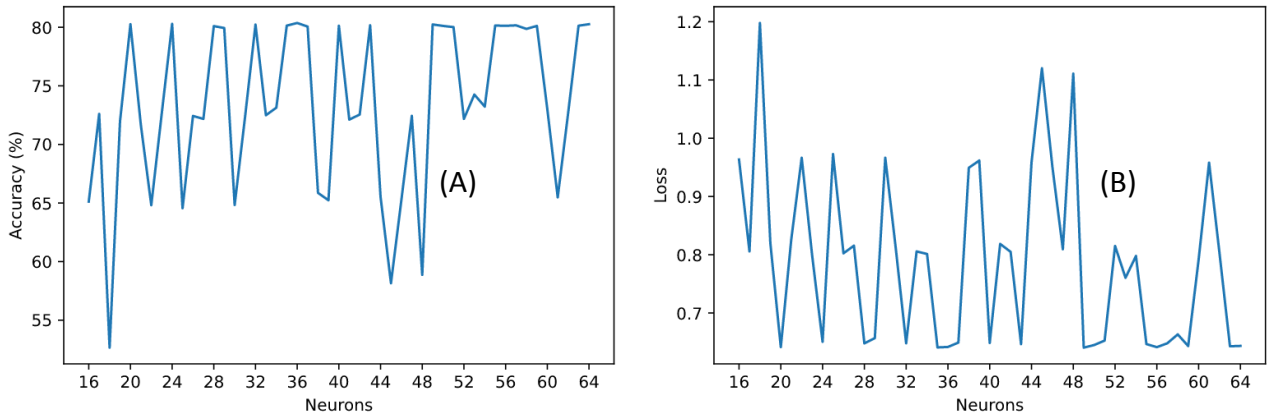

Figure 4.9 – Average accuracy, A), and loss, B), of the created models according to the number of neurons per layer.

After this trial, and making use of the information displayed in Table A1, we considered the structure that has 32 neurons per layer to be optimal for this particular issue and dataset. This was chosen as we desire to maximise average accuracy and Cohen's kappa while simultaneously minimizing average loss and NN complexity.

At this stage, with the intent of further reducing the models' overfitting to the data, we began exploring regularisation methods, starting with dropout.

Usual values for dropout range from around 20% to 80% with some studies suggesting that probability values of 50% should maximize the regularisation effect [123]. Using this information as a target, we added this parameter to the previously chosen NN architecture and recorded its performance (Figure 4.10).

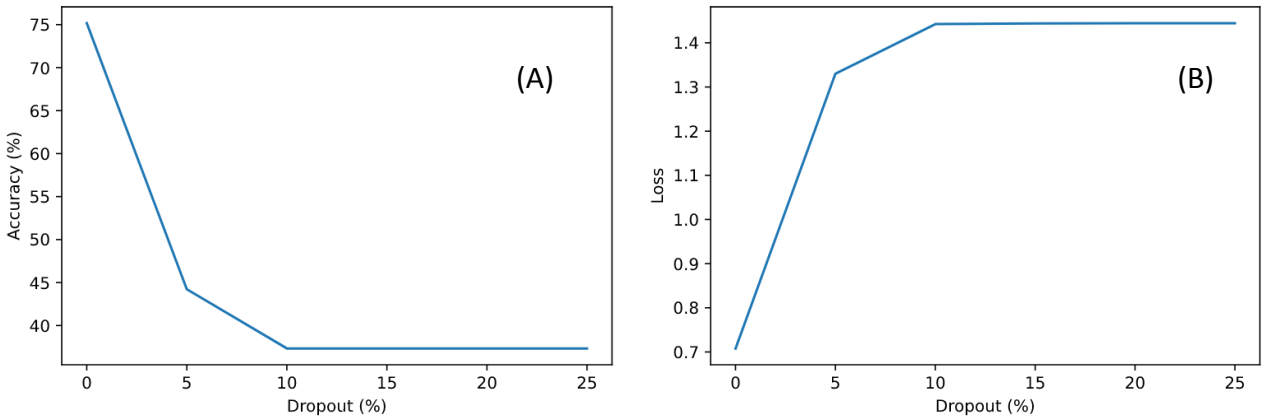

Figure 4.10 – Correlation between average model accuracy, A), or loss, B), and dropout probability.

During training, however, we noticed that the introduction of dropout severely lowers the performance of the created models (as can be seen in Figure 4.10 or Table A2), even in low dropout probabilities. As this goes against the consensus of the influence of dropout on model performance, we hypothesise that this is because the model is already too simple for it to benefit from dropout and, as such, decided to test this parameter again, but now proportionally increasing the number of neurons each layer has according to the probability of dropout (in such a way that, for example, a model with 50% dropout would have 50% more neurons per layer). This would reduce the benefits of model simplification obtained by applying dropout, but should still make the models more robust performance-wise, and, as such, worthwhile to test (Figure 4.11).

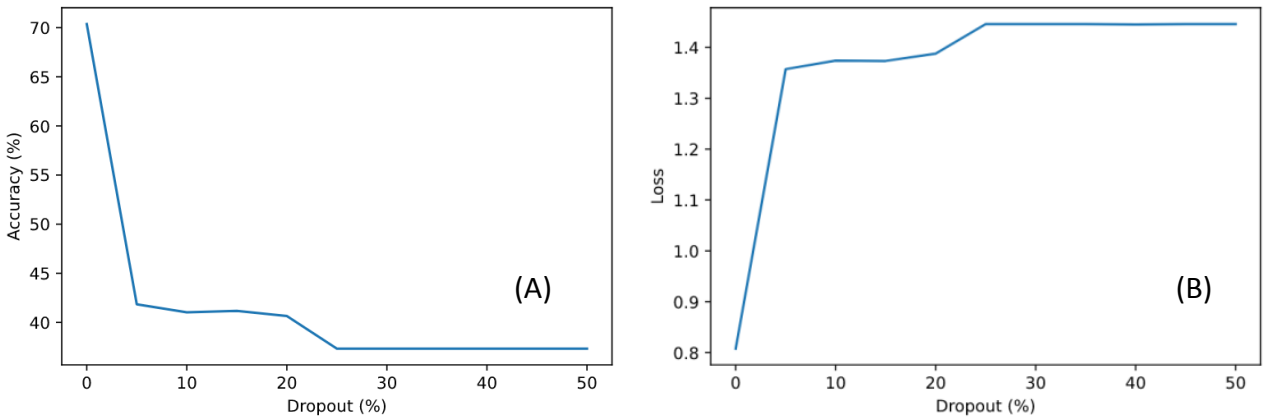

Figure 4.11 - Correlation between average model accuracy, A), or loss, B), and dropout probability, with neuron number adjustment in accordance with dropout probability.

Despite this increase in neuron number, the overall effect of the addition of dropout on the model's average performance is still negative. As it is possible to see in Table A2, the presence of any dropout lowers the average accuracy of the models, while simultaneously significantly reducing all the calculated standard deviations. One thing to note is that the maximum accuracy attained with these models is only substantially reduced when dropout probability is above 20% (raising the threshold from 10% on the previous attempt at implementing dropout), perhaps making it more likely that this reduction in performance is due to the model already being simple enough, and that any further reduction in complexity through this mechanism leads to an inability of the model to properly learn from the data.

This is further confirmed by the fact that the models do not seem to improve over the training epochs, which can be seen in Figure 4.12, in which the epochs, where the best performing model for a specific value of dropout obtained its highest accuracy, seem to lower as dropout increases. This should not be the case, as higher dropout should increase the time necessary before the models' performance converge to the consistent state [123]. This, once again, suggests that this lower accuracy is not due to an underfitting to the data because of a short training time, but because of the inadequacy of the resulting NN architecture.

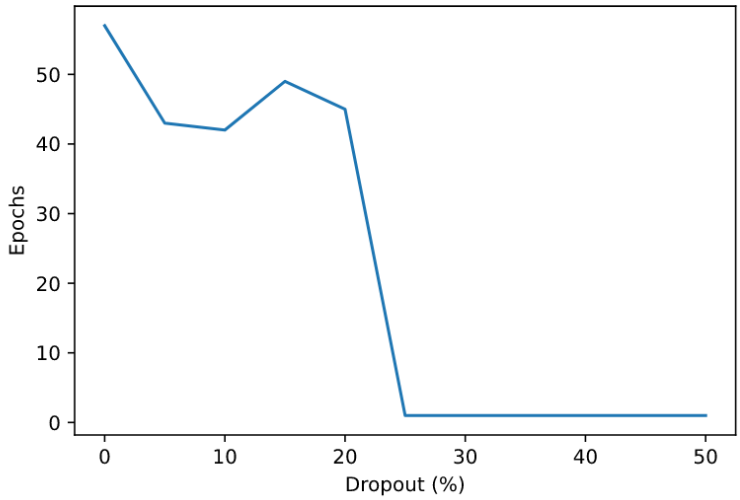

Figure 4.12 – Number of the epoch where the best performing model for a certain dropout probability achieved its highest accuracy.

As previously mentioned, despite the inefficacy of dropout, there are still other regularisation techniques that might prove useful. We began by testing L1 regularisation (Figure 4.13).

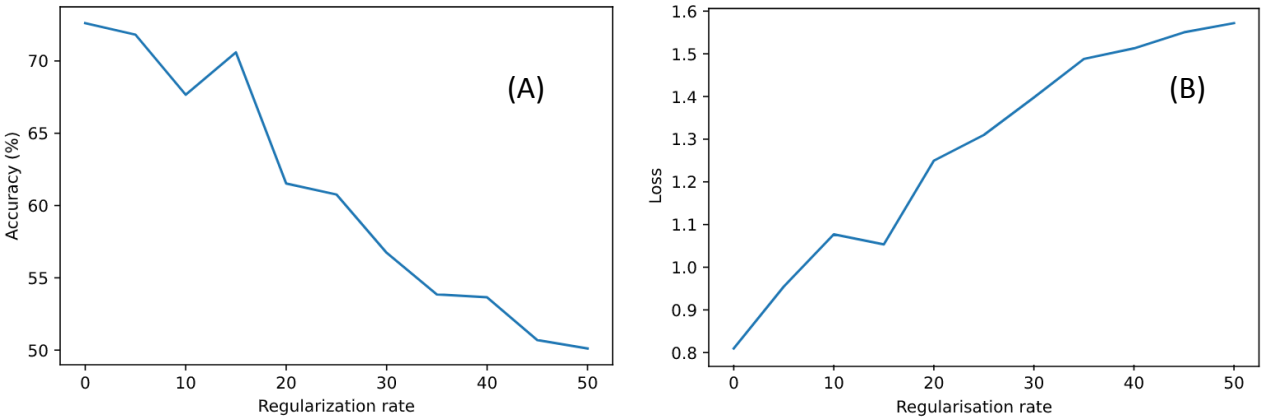

Figure 4.13 - Correlation between average model accuracy, A), or loss, B), and L1 regularisation rate.

Similar to the results obtained when using dropout, the addition of any L1 regularisation seems to significantly reduce the average performance of the created models, with there being a clear trend where the higher the regularisation rate is (and therefore the impact of this regularisation), the lower the efficacy of the model is. This drop in performance is more gradual than what occurred when we added dropout to the neural layers (more details about this behaviour in Table A3). An important characteristic to take note of is that the use of this type of regularisation does not seem to reduce any of the recorded standard deviations (except when it is above 0.3, which might be correlated to the lower recorded metrics), instead increasing this value when compared to not using any L1 regularisation rate. In fact, this effect was especially noteworthy during testing, to the extent that, in order to reliably obtain the same trends for each of the points in these graphs or tables, we had to gather information about the performance of 50 identical NNs, instead of the 10 we normally used for testing other modifications.

As mentioned previously, since L1 regularisation is essentially a feature selection technique, and due to the way features were originally carefully selected to be extracted from the signal, this result from the use of this type of regularisation is not unexpected.

After this, we proceeded to test the effect of L2 regularisation in the same NN architecture (Figure 4.14).

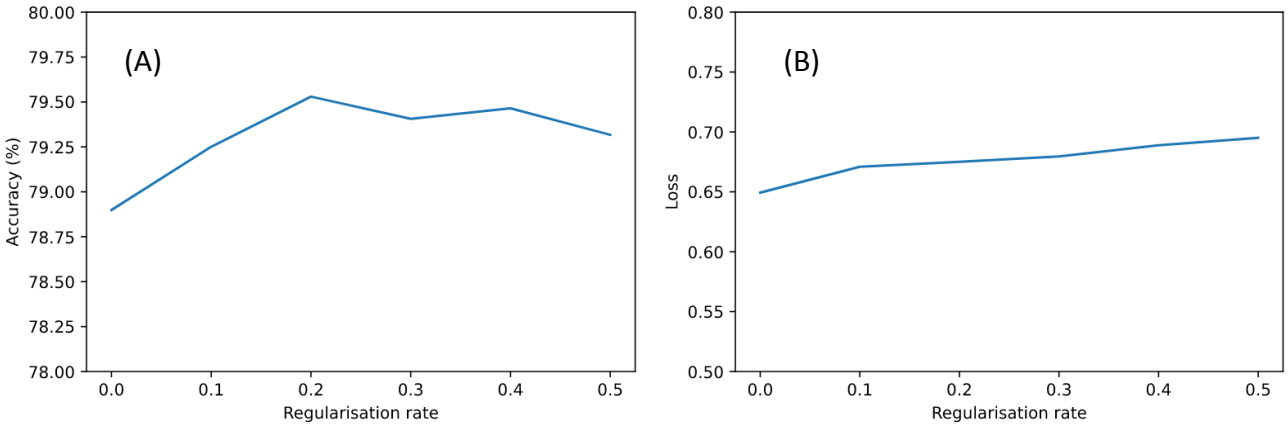

Figure 4.14 – Correlation between average model accuracy, A), or loss, B), and L2 regularisation rate.

As can be observed from Figure 4.14, in this case, L2 regularisation slightly improves average model accuracy, loss, and Cohen's kappa, as well as (represented in Table A4) reducing their standard deviation. Beyond these results are also the expected positive effects of model simplification by weight reduction, as these models should be able to better generalize to previously unseen data.

Expanding this test further, we can find the threshold for the rate when the models start being negatively affected by L2 (Figure 4.15).

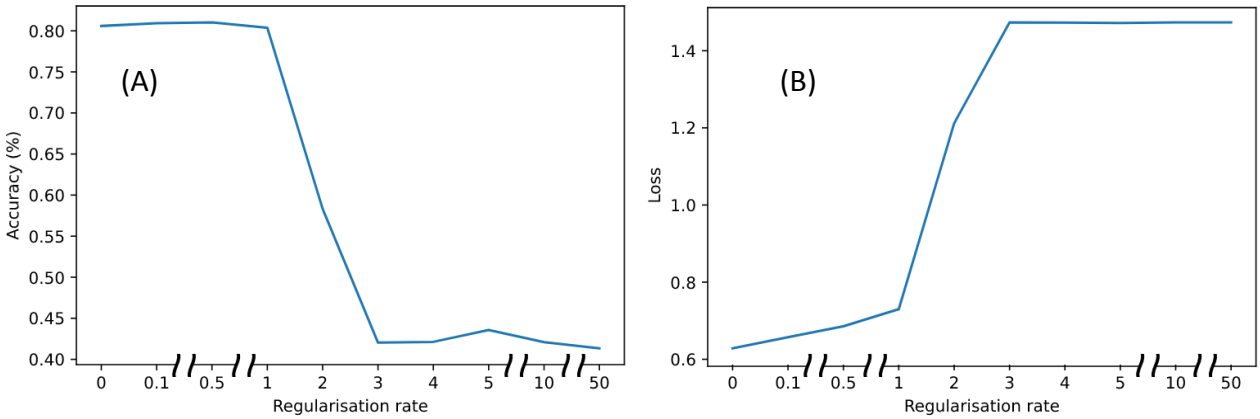

Figure 4.15 – Expanded test on the correlation between average model accuracy, A), or loss, B), and L2 regularisation rate.

From the study of these graphs, and acknowledging that the maximum accuracy of the models only seems to drop from a regularisation rate of above two (which can be observed in Table A5), it seems that as long as the chosen rate is under 2 the results from this addition should be acceptable.

Reaching this point, we gathered what we have learnt so far about the optimal parameters of NN development for this dataset, and proceeded to the development of a finalized version of a *Multilayer Perceptron*. The results obtained are displayed in Table 4.4.

Table 4.4- Values of the chosen metrics for the *Multilayer Perceptron* model.

	Balanced dataset		Raw, unbalanced files		
	Accuracy (%)	Cohen's kappa	Cohen's kappa	Macro average F1-Score (%)	Lowest class accuracy (%)
MLP	80.50	0.7563	0.7586	77.38	52.95 (class 1)

4.3. Model comparison

After comparing NN and non-NN models within their type, it is possible to compare the best models resulting from each of these groups.

Looking into the performance displayed by these models (Figure 4.16, and Tables 4.2 and 4.4) we see that their performance is similar on the test datasets, the *Gradient Boosting* model has slightly higher accuracy (+ 1.84 %) and Cohen's kappa (+ 0.02) on the balanced test set and the M*ultilayer Perceptron* has a higher Cohen's kappa (+ 0.06), and F1-score (+ 1.98%) on the unbalanced test set. Additionally, reusing the proposed cost matrix (Table 4.3) leads us to a very similar cost as well (with the NN having a 1.53% higher associated cost).

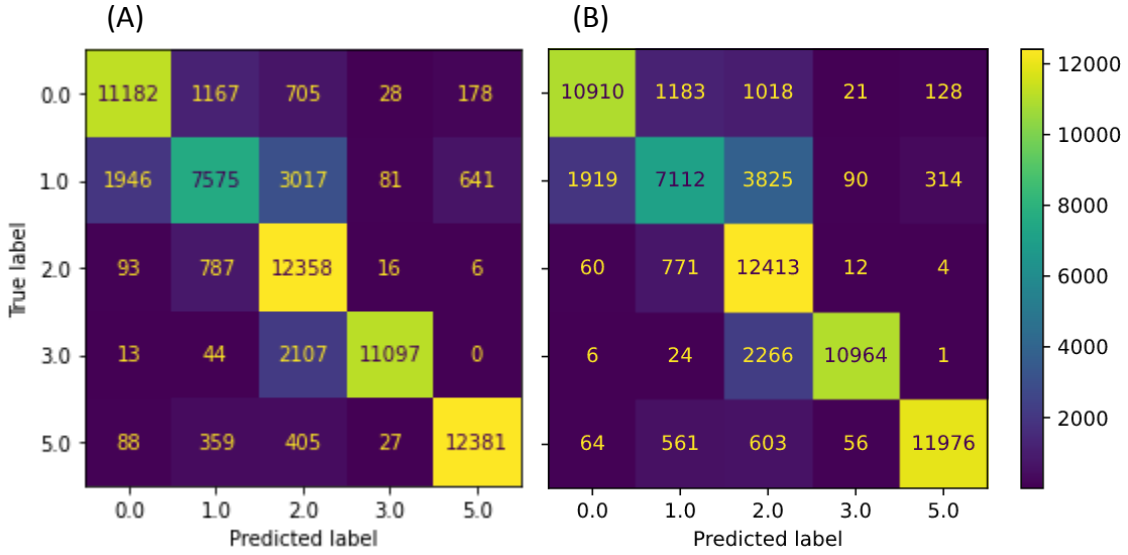

Figure 4.16 – Confusion matrix of the *Gradient Boosting* model, A), and the *Multilayer Perceptron* model, B), for the balanced test set.

One meaningful difference though, is the accuracy of the models for each of the classes in the unbalanced dataset (Figure 4.17).

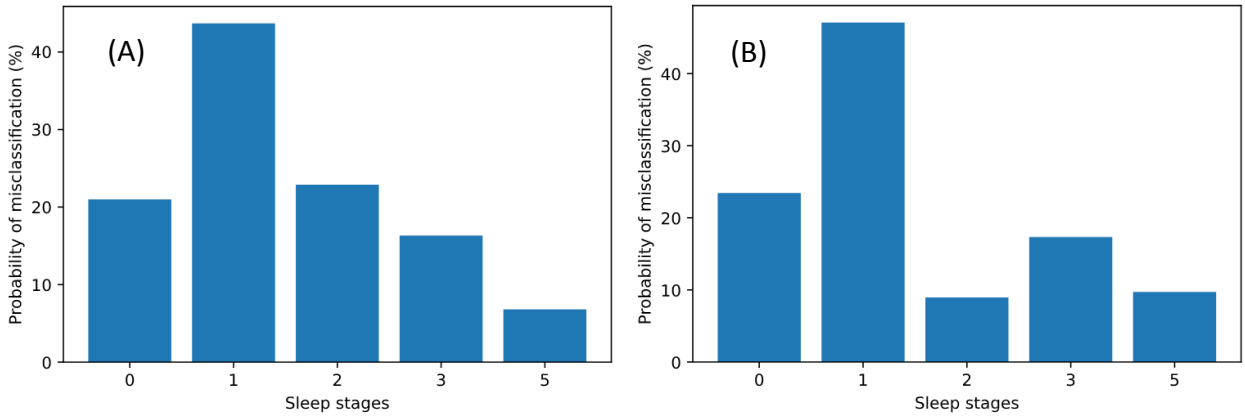

Figure 4.17 – Probability of misclassification for each sleep stage, for the *Gradient Boosting* model, A), and the *Multilayer Perceptron*, B).

Although the accuracy of the NN is inferior for most stages, it still performs better on stage 2. This was possible to see even on the balanced dataset, however, this difference seems to have further intensified in the complete unbalanced dataset. We believe that this difference in performance for stage 2 is due to the NN's increased ability to generalize to data it has not been trained with when compared to the non-NNs developed throughout this work. Usually, such a difference in performance might not be significant (especially if it is only noticeable once tested with a larger set of data), however, as mentioned previously, stage 2 is one of the most common stages during normal sleep (representing 32.82% of the intervals in our dataset), and, as such, any slight variation in its classification represents a disproportionate impact in the performance of the model in real-world circumstances.

Because of this, and due to their similarity in terms of the recorded metrics, we propose that it is more adequate to utilise the developed NN than the other models.

Using this, we can now more directly compare the performance of our model with others that used information from the same dataset (MESA). In this regard, the results we have obtained (80.50% accuracy, 0.7586 Cohen's kappa, and 77.38% macro-average F1-score) can be considered as good. Kudo et al. (2022), using information from PPGs and accelerometers for classification (extracting this data from the public datasets Apple watch Sleep dataset [124], and MESA), achieved a macro F1 score of 0.655 and Cohen's kappa score of 0.527, on their GRU-based recurrent neural network [125]. Another similar study, published by Sridhar et al. (2020), utilizing a fully convolutional neural network with dilated convolutional blocks and using the ECG signal of both the Sleep Heart Health Study and the MESA dataset for training, validation and testing of the developed algorithm,

obtained an overall performance of 77% accuracy and 0.66 Cohen's kappa, for four-stage classification, against the reference stages on a held-out portion of the datasets used for training [120].

These results are despite the fact that during this work we exclusively used the PPG signal for classifying the sleep stages, which, notwithstanding its convenience in terms of integration with wearable devices, tends to be contaminated with noise artefacts [126] and is just one of the many signals used during a normal PSG.

Our algorithm also matches up positively with other recent studies that only use PPG, presenting higher accuracy and Cohen's kappa for sleep stage classification of 4 different classes [17, 127].

4.4. Real-world Corroboration of Results

Due to the previously displayed results, the *Multilayer Perceptron* model was chosen to be compared to the results obtained from "Sleep as Android". As a result of this comparison, we acquired the data displayed in Figure 4.18.

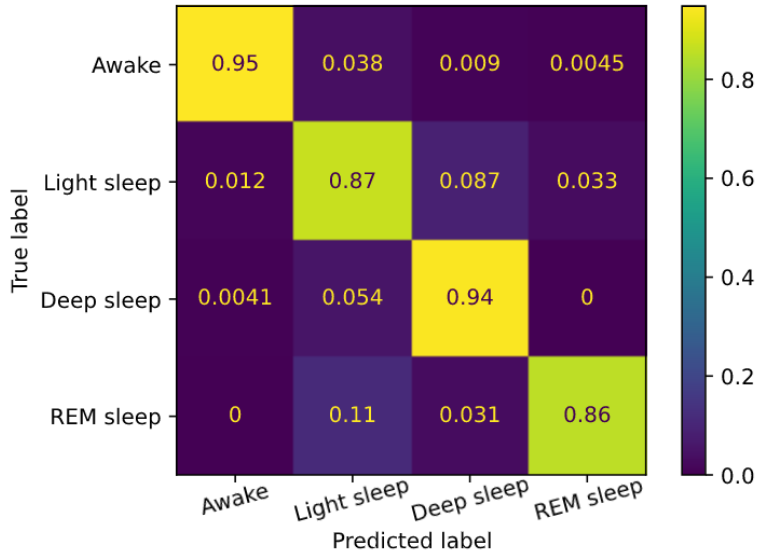

Figure 4.18 – Normalized confusion matrix of the results obtained from the *Multilayer Perceptron* for the "Sleep as Android" data.

The results obtained (Table 4.5) present a strong level of agreement [109]. These are valuable results as, first of all, "Sleep as Android" is one of the most reviewed android sleep analysis smartphone applications [128], with over ten million downloads [129, 130] (its paid version having over a thousand [131, 132]), showing that there is a market for algorithms with the sort of accuracy our models have. Second, as previously described, the devices and setting used for this test are different from the ones used for training the

algorithm, exhibiting our model's ability to work in a wider set of circumstances (more specifically in a real-world situation). Third, it assists in confirming the reliability of the developed algorithm, as "Sleep as Android" estimates sleep stages mainly utilizing actigraphy, which is sometimes considered as an alternative to PSG [62, 133, 134], with their own studies agreeing with these results [135].

Table 4.5- Values of the chosen metrics for the *Multilayer Perceptron* model on the acquired data.

Accuracy (%)	Cohen's kappa	Macro average F1-Score (%)	Lowest Class Agreement (%)
90.96	0.8663	90.52	85.55 (class REM)

5. Conclusion

5.1. General Results

This thesis' main objective was the development of a ML algorithm that focuses on the detection and classification of sleep cycles.

At the start, we had to select which signals to utilise. As PPG is very convenient, inexpensive, and easily integrated into portable devices, it is a promising signal to use for sleep stage classification based on wearables. After that, since we intended to build NNs that require a significant amount of data to be trained, we had to find databases that not only contained PPG signals and information about sleep stages, but that were also vast enough. The MESA dataset [73–75] was selected, as it has 2056 full overnight polysomnographies and contains PPG signals, thus being appropriate for our ends.

Pre-processing and feature selection were the next step, and here we benefitted from the current state-of-the-art. Although, typically, this type of sleep study is done through the use of more than just PPG, we were nevertheless able to build optimal filters and select features for this purpose, such as signal characteristics related to HRV, which, while often connected to studies utilising ECG, proved useful for our work.

In regard to the ML models, we created both NN and non-NN models. For the non-NNs, the best performing model (*Gradient Boosting*) presented 82.34% accuracy, a Cohen's kappa of 0.7792, and a macro average F1-Score of 75.40% on the complete, unbalanced test set. During the training of these models, we concluded that, beyond the normal effects of overfitting, the increase of data from a single individual is deleterious to the model's ability to generalize its predictions to other individuals, despite, in theory, this being beneficial to algorithm development.

For the NNs, the best performing model (*Multilayer Perceptron*) presented 80.50% accuracy, 0.7563 Cohen's kappa, and a macro average F1-Score of 77.38% on the complete, unbalanced test set. After an extensive search for the optimal configuration of hyperparameters for the NN, we found that the model consistently performed better in a 3 hidden layer, 32 neurons per layer, structure, with all hidden layers having a L2 regularisation rate of 0.4. Overall, we found that performance tends to be highest for models with 3 or 7 layers, with it dropping sharply outside these limits. Similarly for neuron count, accuracy dropped to around 20% for any number of neurons per layer outside of the interval between 16 and 64, where it seems mostly stable at around 80% accuracy and optimal at 32 neurons per layer.

We also tested some other commonly used regularisation methods, such as L1, L2, or dropout. In the case of the latter, despite usually being described as improving model performance, it failed to do so in this case, instead leading to a decrease in performance (even only 5% dropout lowers average accuracy to 41.84%). This decrease seems tied to dropout probability, where the higher the probability, the worse the performance is. This effect reaches a plateau at 37.35% accuracy, at which point all created models seem to have very homogeneous characteristics. The addition of L1 regularisation also seems to be detrimental to model development, with the higher the rate, the worse its impact on the model's accuracy. On the other hand, L2 regularisation seems to improve the effectiveness of the models, and, while we found a regularisation rate of 0.4 to be optimal, there seems to be a wide range of values (from 0.1 to 2) where the model still benefits from its addition.

These results are promising as, while some models are able to achieve higher accuracy [48, 68], they do so while using more signals (usually EEG, EOG, or ECG), which significantly restricts their usability for everyday applications. Conversely, we reached better performance than many other models, including recently published studies that make use of more signals or features [119], or employ the same dataset [120, 125].

One of the goals of this work was to test the developed model's performance in a real-world scenario. To achieve this, we simultaneously recorded data using a biosignalsPlux device with PPG and accelerometer sensors and a widely used Android sleep scoring application ("Sleep as Android") paired with a commercially available wearable device (TicWatch E2).

Associated with the device developed by PLUX was one of our previously created models (more specifically, the *Multilayer Perceptron*). This validation was done to ensure its performance was not tied exclusively to data from the chosen dataset and to compare its effectiveness with other popular sleep algorithms. The selection of the *Multilayer Perceptron* was justified by the fact that, despite its performance being slightly inferior for the balanced dataset when compared to the *Gradient Boosting*, there are significant differences in accuracy (with the NN having a 13.94% higher accuracy) for stage 2 sleep classification, which represents 32.82% of the data points we used, in a larger test set. As for the corroboration of the previously attained results by resorting to a comparison to the scoring of our own data acquisitions performed by another algorithm (associated with the app "Sleep as Android"), we obtained a strong level of agreement (90.96% accuracy, 0.8663 Cohen's kappa and a macro average F1-Score of 90.52%). This leads us to believe in the potential of the developed algorithm to be used in real-world scenarios.

5.2. Future Work

While we fulfilled the main goals of this work, it still presents some limitations that could be improved, namely in terms of feature acquisition and extraction.

Future studies should attempt to integrate these algorithms into devices. This way, not only is it possible to increase the similarity between the devices and algorithms being compared, but it should also be easier to acquire a larger amount of data, ideally, from a larger set of individuals as well.

The recording of more data itself would also likely lead to improvements in the determination of the real-world performance of the models, besides the potential use of this data for model training. In this regard, the recording and comparison of results with a PSG study would be optimal.

Additionally, during feature extraction, we chose to reduce the number and quality of the entropies used as features, due to time and computation constraints. As, even after this, these were some of the most relevant features, the extraction and use of them without averaging the signal beforehand could lead to some performance improvements.

Finally, throughout this work several models were created, some of them having similar levels of accuracy and other selected metrics to the final model developed. Due to this, a complementary study that could be done is the creation of another ensemble model that utilises the output of these models as inputs, as these types of models tend to have a better performance than the sum of their parts [136].

Bibliography

[1] J. Lourenço, “The NOVAthesis LATEX Template User’s Manual.”, *Nova University Lisbon*. https://github.com/joaomlourenco/novathesis_word (accessed Aug. 26, 2022).

[2] J. D. Minkel *et al.*, “Sleep deprivation and stressors: Evidence for elevated negative affect in response to mild stressors when sleep deprived”, *Emotion*, vol. 12, no. 5, pp. 1015–1020, 2012, doi: 10.1037/a0026871.

[3] S. L. Worley, “The extraordinary importance of sleep: The detrimental effects of inadequate sleep on health and public safety drive an explosion of sleep research”, *P T*, vol. 43, no. 12, pp. 758–763, 2018.

[4] D. F. Dinges, “An overview of sleepiness and accidents”, *Journal of Sleep Research*, vol. 4. pp. 4–14, 1995. doi: 10.1111/j.1365-2869.1995.tb00220.x.

[5] D. S. Lauderdale, K. L. Knutson, L. L. Yan, K. Liu, and P. J. Rathouz, “Self-reported and measured sleep duration: How similar are they?”, *Epidemiology*, vol. 19, no. 6, pp. 838–845, 2008, doi: 10.1097/EDE.0b013e318187a7b0.

[6] O. Tkachenko and D. F. Dinges, “Interindividual variability in neurobehavioral response to sleep loss: A comprehensive review”, *Neurosci. Biobehav. Rev.*, vol. 89, no. October 2017, pp. 29–48, 2018, doi: 10.1016/j.neubiorev.2018.03.017.

[7] J. W. Shepard *et al.*, “History of the development of sleep medicine in the United States.”, *J. Clin. Sleep Med.*, vol. 1, no. 1, pp. 61–82, 2005, doi: 10.5664/jcsm.26298.

[8] RazerM, “File:Sleep Hypnogram.svg - Wikimedia Commons”, Jan. 29, 2011. https://commons.wikimedia.org/wiki/File:Sleep_Hypnogram.svg (accessed Aug. 19, 2022).

[9] H. R. Colten and B. M. Altevogt, *Sleep disorders and sleep deprivation: An unmet public health problem*. 2006. doi: 10.17226/11617.

[10] I. Feinberg and T. C. Floyd, “Systematic trends across the night in human sleep cycles.”, *Psychophysiology*, vol. 16, no. 3, pp. 283–91, May 1979, doi: 10.1111/j.1469-8986.1979.tb02991.x.

[11] P. Memar and F. Faradji, “A Novel Multi-Class EEG-Based Sleep Stage Classification System.”, *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 1, pp. 84–95, 2018, doi: 10.1109/TNSRE.2017.2776149.

[12] J. Malik, Y.-L. Lo, and H.-T. Wu, “Sleep-wake classification via quantifying heart rate variability by convolutional neural network.”, *Physiol. Meas.*, vol. 39, no. 8, p. 085004, 2018, doi: 10.1088/1361-6579/aad5a9.

[13] NascarEd, “File:Sleep Stage N3.png - Wikimedia Commons”, Feb. 07, 2013. https://commons.wikimedia.org/wiki/File:Sleep_Stage_N3.png (accessed Aug. 19, 2022).

[14] A. L. Loomis, E. N. Harvey, and G. A. Hobart, “Cerebral states during sleep, as studied by human brain potentials.”, *J. Exp. Psychol.*, vol. 21, no. 2, pp. 127–144, 1937, doi: 10.1037/h0057431.

[15] W. Dement and N. Kleitman, “Cyclic variations in EEG during sleep and their relation to eye movements, body motility, and dreaming.”, *Electroencephalogr. Clin. Neurophysiol.*, vol. 9, no. 4, pp. 673–90, Nov. 1957, doi: 10.1016/0013-4694(57)90088-3.

[16] F. Riganello, V. Prada, A. Soddu, C. Di Perri, and W. G. Sannita, “Circadian Rhythms and

Measures of CNS/Autonomic Interaction", *Int. J. Environ. Res. Public Health*, vol. 16, no. 13, Jul. 2019, doi: 10.3390/IJERPH16132336.

[17] P. Fonseca *et al.*, "Validation of Photoplethysmography-Based Sleep Staging Compared With Polysomnography in Healthy Middle-Aged Adults.", *Sleep*, vol. 40, no. 7, 2017, doi: 10.1093/sleep/zsx097.

[18] P. Fonseca, X. Long, M. Radha, R. Haakma, R. M. Aarts, and J. Rolink, "Sleep stage classification with ECG and respiratory effort.", *Physiol. Meas.*, vol. 36, no. 10, pp. 2027–40, Oct. 2015, doi: 10.1088/0967-3334/36/10/2027.

[19] V. K. Somers, M. E. Dyken, A. L. Mark, and F. M. Abboud, "Sympathetic-nerve activity during sleep in normal subjects.", *N. Engl. J. Med.*, vol. 328, no. 5, pp. 303–7, Feb. 1993, doi: 10.1056/NEJM199302043280502.

[20] A. M. Fink, U. G. Bronas, and M. W. Calik, "Autonomic regulation during sleep and wakefulness: a review with implications for defining the pathophysiology of neurological disorders.", *Clin. Auton. Res.*, vol. 28, no. 6, pp. 509–518, 2018, doi: 10.1007/s10286-018-0560-9.

[21] J. V. Rundo and R. Downey, "Polysomnography", 2019, pp. 381–392. doi: 10.1016/B978-0-444-64032-1.00025-4.

[22] W. Karlen, C. Mattiussi, and D. Floreano, "Sleep and wake classification with ECG and respiratory effort signals", *IEEE Trans. Biomed. Circuits Syst.*, vol. 3, no. 2, pp. 71–78, 2009, doi: 10.1109/TBCAS.2008.2008817.

[23] J. M. Kelly, R. E. Strecker, and M. T. Bianchi, "Recent Developments in Home Sleep-Monitoring Devices", *ISRN Neurol.*, vol. 2012, pp. 1–10, 2012, doi: 10.5402/2012/768794.

[24] M. R. Pioli, A. M. V. Ritter, A. P. de Faria, and R. Modolo, "White coat syndrome and its variations: differences and clinical impact", *Integr. Blood Press. Control*, vol. 11, p. 73, 2018, doi: 10.2147/IBPC.S152761.

[25] M. De Zambotti, N. Cellini, A. Goldstone, I. M. Colrain, and F. C. Baker, "Wearable Sleep Technology in Clinical and Research Settings", *Med. Sci. Sports Exerc.*, vol. 51, no. 7, pp. 1538–1557, 2019, doi: 10.1249/MSS.0000000000001947.

[26] N. A. Collop *et al.*, "Obstructive sleep apnea devices for Out-Of-Center (OOC) testing: Technology evaluation", *J. Clin. Sleep Med.*, vol. 7, no. 5, pp. 531–548, 2011, doi: 10.5664/JCSM.1328.

[27] M. de Zambotti, L. Menghini, N. Cellini, C. Goldstein, and F. C. Baker, "Performance of consumer wearable sleep technology", in *Reference Module in Neuroscience and Biobehavioral Psychology*, Elsevier, 2021. doi: 10.1016/B978-0-12-822963-7.00199-7.

[28] L. Fiorillo *et al.*, "Automated sleep scoring: A review of the latest approaches", *Sleep Med. Rev.*, vol. 48, p. 101204, 2019, doi: 10.1016/j.smrv.2019.07.007.

[29] M. Younes, J. Raneri, and P. Hanly, "Staging Sleep in Polysomnograms: Analysis of Inter-Scorer Variability", *J. Clin. Sleep Med.*, vol. 12, no. 6, p. 885, 2016, doi: 10.5664/JCSM.5894.

[30] N. A. Collop, "Scoring variability between polysomnography technologists in different sleep laboratories", *Sleep Med.*, vol. 3, no. 1, pp. 43–47, 2002, doi: 10.1016/S1389-9457(01)00115-0.

[31] M. K. Pavlova and V. Latreille, "Sleep Disorders", *Am. J. Med.*, vol. 132, no. 3, pp. 292–299, 2019, doi: 10.1016/j.amjmed.2018.09.021.

[32] C. A. Kushida *et al.*, “Practice parameters for the indications for polysomnography and related procedures: An update for 2005”, *Sleep*, vol. 28, no. 4, pp. 499–521, 2005, doi: 10.1093/sleep/28.4.499.

[33] T. Sheerman-Chase, “‘Volunteer Duty’ Psychology Testing”, Oct. 29, 2012. https://www.flickr.com/photos/tim_uk/8135755109/ (accessed Aug. 16, 2022).

[34] J. Haba-Rubio and J. Krieger, “Evaluation Instruments for Sleep Disorders: A Brief History of Polysomnography and Sleep Medicine”, *Intell. Syst. Control Autom. Sci. Eng.*, vol. 64, pp. 19–31, 2012, doi: 10.1007/978-94-007-5470-6_2.

[35] I. M. Colrain, “The K-complex: a 7-decade history”, *Sleep*, vol. 28, no. 2, pp. 255–273, Feb. 2005, doi: 10.1093/SLEEP/28.2.255.

[36] A. L. Loomis, E. N. Harvey, and G. Hobart, “POTENTIAL RHYTHMS OF THE CEREBRAL CORTEX DURING SLEEP”, *Science*, vol. 81, no. 2111, pp. 597–598, 1935, doi: 10.1126/SCIENCE.81.2111.597.

[37] A. L. Loomis, E. N. Harvey, and G. Hobart, “FURTHER OBSERVATIONS ON THE POTENTIAL RHYTHMS OF THE CEREBRAL CORTEX DURING SLEEP”, *Science*, vol. 82, no. 2122, pp. 198–200, 1935, doi: 10.1126/SCIENCE.82.2122.198.

[38] H. Davis, P. A. Davis, A. L. Loomis, E. N. Harvey, and G. Hobart, “CHANGES IN HUMAN BRAIN POTENTIALS DURING THE ONSET OF SLEEP”, *Science*, vol. 86, no. 2237, pp. 448–450, 1937, doi: 10.1126/SCIENCE.86.2237.448.

[39] A. L. Loomis, E. N. Harvey, and I. Garret A. Hobart, “DISTRIBUTION OF DISTURBANCE-PATTERNS IN THE HUMAN ELECTROENCEPHALOGRAM, WITH SPECIAL REFERENCE TO SLEEP”, *https://doi.org/10.1152/jn.1938.1.5.413*, vol. 1, no. 5, pp. 413–430, Sep. 1938, doi: 10.1152/JN.1938.1.5.413.

[40] H. Davis, P. A. Davis, A. L. Loomis, E. N. Harvey, and G. Hobart, “ELECTRICAL REACTIONS OF THE HUMAN BRAIN TO AUDITORY STIMULATION DURING SLEEP”, *https://doi.org/10.1152/jn.1939.2.6.500*, vol. 2, no. 6, pp. 500–514, Nov. 1939, doi: 10.1152/JN.1939.2.6.500.

[41] M. Deak and L. J. Epstein, “The History of Polysomnography”, *Sleep Med. Clin.*, vol. 4, no. 3, pp. 313–321, Sep. 2009, doi: 10.1016/J.JSMC.2009.04.001.

[42] C. Iber, *The AASM manual for the scoring of sleep and associated events : rules, terminology and technical specifications*. Westchester IL: American Academy of Sleep Medicine, 2007.

[43] “AASM Scoring Manual - American Academy of Sleep Medicine.” https://aasm.org/clinical-resources/scoring-manual/ (accessed Sep. 02, 2022).

[44] A. Kales Rechtschaffen, Allan., University of California, Los Angeles., Brain Information Service., National Institute of Neurological Diseases and Blindness (U.S.), *A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects*. Washington, DC: United States Government Printing Office, 1968.

[45] J. V. Rundo and R. Downey, “Polysomnography”, *Handb. Clin. Neurol.*, vol. 160, pp. 381–392, Jan. 2019, doi: 10.1016/B978-0-444-64032-1.00025-4.

[46] J. M. Beecroft, M. Ward, M. Younes, S. Crombach, O. Smith, and P. J. Hanly, “Sleep monitoring in the intensive care unit: Comparison of nurse assessment, actigraphy and polysomnography”, *Intensive Care Med.*, vol. 34, no. 11, pp. 2076–2083, 2008, doi: 10.1007/s00134-008-1180-y.

[47] D. C. Lim *et al.*, “Reinventing polysomnography in the age of precision medicine”, *Sleep*

Med. Rev., vol. 52, no. January, p. 101313, 2020, doi: 10.1016/j.smrv.2020.101313.

[48] O. Yildirim, U. B. Baloglu, and U. R. Acharya, "A deep learning model for automated sleep stages classification using PSG signals", *Int. J. Environ. Res. Public Health*, vol. 16, no. 4, 2019, doi: 10.3390/ijerph16040599.

[49] A. M. Tautan, A. C. Rossi, R. De Francisco, and B. Ionescu, "Automatic Sleep Stage Detection: A Study on the Influence of Various PSG Input Signals", *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2020-July, pp. 5330–5334, 2020, doi: 10.1109/EMBC44109.2020.9175628.

[50] A. K. Sabil *et al.*, "Automatic identification of sleep and wakefulness using single-channel EEG and respiratory polygraphy signals for the diagnosis of obstructive sleep apnea", *J. Sleep Res.*, vol. 28, no. 2, 2019, doi: 10.1111/jsr.12795.

[51] V. Muto *et al.*, "0315 Inter- And Intra-expert Variability In Sleep Scoring: Comparison Between Visual And Automatic Analysis", *Sleep*, vol. 41, no. suppl_1, pp. A121–A121, Apr. 2018, doi: 10.1093/sleep/zsy061.314.

[52] R. B. Berry, G. Hill, L. Thompson, and V. McLaurin, "Portable monitoring and autotitration versus polysomnography for the diagnosis and treatment of sleep apnea", *Sleep*, vol. 31, no. 10, pp. 1423–1431, 2008, doi: 10.5665/sleep/31.10.1423.

[53] M. A. Akkaş, R. SOKULLU, and H. Ertürk Çetin, "Healthcare and patient monitoring using IoT", *Internet of Things*, vol. 11, p. 100173, Sep. 2020, doi: 10.1016/J.IOT.2020.100173.

[54] M. E. H. Chowdhury, A. Khandakar, Y. Qiblawey, M. B. I. Reaz, M. T. Islam, and F. Touati, "Machine Learning in Wearable Biomedical Systems", *Sport. Sci. Hum. Heal. - Differ. Approaches*, Aug. 2020, doi: 10.5772/INTECHOPEN.93228.

[55] A. Mahajan, G. Pottie, and W. Kaiser, "Transformation in Healthcare by Wearable Devices for Diagnostics and Guidance of Treatment", *ACM Trans. Comput. Healthc.* , vol. 1, no. 1, Mar. 2020, doi: 10.1145/3361561.

[56] T. Morgenthaler *et al.*, "Practice parameters for the use of actigraphy in the assessment of sleep and sleep disorders: An update for 2007", *Sleep*, vol. 30, no. 4, pp. 519–529, 2007, doi: 10.1093/sleep/30.4.519.

[57] N. Thonte, "Smartway - Free Image by Nilesh Thonte on PixaHive.com", Nov. 26, 2020. https://pixahive.com/photo/smartway/ (accessed Aug. 20, 2022).

[58] S. A. Imtiaz, "A systematic review of sensing technologies for wearable sleep staging", *Sensors*, vol. 21, no. 5, pp. 1–21, 2021, doi: 10.3390/s21051562.

[59] Z. Hussain, Q. Z. Sheng, W. E. Zhang, J. Ortiz, and S. Pouriyeh, "A Review of the Non-Invasive Techniques for Monitoring Different Aspects of Sleep", pp. 1–19, 2021, [Online]. Available: http://arxiv.org/abs/2104.12964

[60] A. T. M. VAN DE WATER, A. HOLMES, and D. A. HURLEY, "Objective measurements of sleep for non-laboratory settings as alternatives to polysomnography - a systematic review", *J. Sleep Res.*, vol. 20, no. 1pt2, pp. 183–200, Mar. 2011, doi: 10.1111/j.1365-2869.2009.00814.x.

[61] B. Stucky *et al.*, "Validation of fitbit charge 2 sleep and heart rate estimates against polysomnographic measures in shift workers: naturalistic study", *J. Med. Internet Res.*, vol. 23, no. 10, pp. 1–20, 2021, doi: 10.2196/26476.

[62] C. McCall and W. V. McCall, "Comparison of actigraphy with polysomnography and sleep logs in depressed insomniacs.", *J. Sleep Res.*, vol. 21, no. 1, pp. 122–7, Feb. 2012, doi: 10.1111/j.1365-2869.2011.00917.x.

[63] A. Sadeh, K. M. Sharkey, and M. A. Carskadon, “Activity-based sleep-wake identification: An empirical test of methodological issues”, *Sleep*, vol. 17, no. 3, pp. 201–207, 1994, doi: 10.1093/sleep/17.3.201.

[64] T Mitchell, B Buchanan, G DeJong, T Dietterich, and P Rosenbloom, and A. Waibel, “Machine Learning”, *http://dx.doi.org/10.1146/annurev.cs.04.060190.002221*, vol. 4, no. 1, pp. 417–433, Nov. 2003, doi: 10.1146/ANNUREV.CS.04.060190.002221.

[65] K. El Bouchefry and R. S. de Souza, “Learning in Big Data: Introduction to Machine Learning”, *Knowl. Discov. Big Data from Astron. Earth Obs. Astrogeoinformatics*, pp. 225–249, Apr. 2020, doi: 10.1016/B978-0-12-819154-5.00023-0.

[66] S. Haghayegh, S. Khoshnevis, M. H. Smolensky, K. R. Diller, and R. J. Castriotta, “Performance comparison of different interpretative algorithms utilized to derive sleep parameters from wrist actigraphy data”, *Chronobiol. Int.*, vol. 36, no. 12, pp. 1752–1760, 2019, doi: 10.1080/07420528.2019.1679826.

[67] A. S. Cakmak *et al.*, “An unbiased, efficient sleep–wake detection algorithm for a population with sleep disorders: Change point decoder”, *Sleep*, vol. 43, no. 8, pp. 1–10, 2020, doi: 10.1093/sleep/zsaa011.

[68] O. Tsinalis, P. M. Matthews, and Y. Guo, “Automatic Sleep Stage Scoring Using Time-Frequency Analysis and Stacked Sparse Autoencoders”, *Ann. Biomed. Eng.*, vol. 44, no. 5, pp. 1587–1597, 2016, doi: 10.1007/s10439-015-1444-y.

[69] H. Ramchoun, M. Amine, J. Idrissi, Y. Ghanou, and M. Ettaouil, “Multilayer Perceptron: Architecture Optimization and Training”, *Int. J. Interact. Multimed. Artif. Intell.*, vol. 4, no. 1, p. 26, 2016, doi: 10.9781/IJIMAI.2016.415.

[70] E. Eğrioğlu, Ç. H. Aladağ, and S. Günay, “A new model selection strategy in artificial neural networks”, *Appl. Math. Comput.*, vol. 195, no. 2, pp. 591–597, Feb. 2008, doi: 10.1016/J.AMC.2007.05.005.

[71] H. Lee *et al.*, “NCH Sleep DataBank: A Large Collection of Real-world Pediatric Sleep Studies”, Feb. 2021, doi: 10.25822/JPDR-VZ50.

[72] S. F. Quan *et al.*, “The Sleep Heart Health Study: Design, rationale, and methods”, *Sleep*, vol. 20, no. 12, pp. 1077–1085, 1997, doi: 10.1093/sleep/20.12.1077.

[73] “Multi-Ethnic Study of Atherosclerosis - Sleep Data - National Sleep Research Resource - NSRR.” https://sleepdata.org/datasets/mesa (accessed Jun. 24, 2022).

[74] G.-Q. Zhang *et al.*, “The National Sleep Research Resource: towards a sleep data commons.”, *J. Am. Med. Inform. Assoc.*, vol. 25, no. 10, pp. 1351–1358, 2018, doi: 10.1093/jamia/ocy064.

[75] X. Chen *et al.*, “Racial/ethnic differences in sleep disturbances: The Multi-Ethnic Study of Atherosclerosis (MESA)”, *Sleep*, vol. 38, no. 6, pp. 877–888, Jun. 2015, doi: 10.5665/sleep.4732.

[76] “Administrative - MESA Variables - Sleep Data - National Sleep Research Resource - NSRR.” https://sleepdata.org/datasets/mesa/variables?folder=Administrative (accessed Jun. 24, 2022).

[77] Y. Liang, M. Elgendi, Z. Chen, and R. Ward, “An optimal filter for short photoplethysmogram signals”, *Sci. Data 2018 51*, vol. 5, no. 1, pp. 1–12, May 2018, doi: 10.1038/sdata.2018.76.

[78] M. Elgendi, “Optimal Signal Quality Index for Photoplethysmogram Signals.”, *Bioeng. (Basel, Switzerland)*, vol. 3, no. 4, Sep. 2016, doi: 10.3390/bioengineering3040021.

[79] J. K. Kim and J. Mok Ahn, “Design of an Optimal Preamplifier of Photoplethysmogram for Age-Related Vascular Stiffening Evaluation”, *Int. J. Eng. Res. Technol.*, vol. 12, no. 10, pp. 1639–1646, 2019, Accessed: Jun. 28, 2022. [Online]. Available: http://www.irphouse.com

[80] K. Pilt, R. Ferenets, K. Meigas, L. G. Lindberg, K. Temitski, and M. Viigimaa, “New Photoplethysmographic Signal Analysis Algorithm for Arterial Stiffness Estimation”, *Sci. World J.*, vol. 2013, 2013, doi: 10.1155/2013/169035.

[81] L. Salahuddin, J. Cho, M. G. Jeong, and D. Kim, “Ultra short term analysis of heart rate variability for monitoring mental stress in mobile settings”, *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Int. Conf.*, vol. 2007, pp. 4656–4659, 2007, doi: 10.1109/IEMBS.2007.4353378.

[82] F. T. Sun, C. Kuo, H. T. Cheng, S. Buthpitiya, P. Collins, and M. Griss, “Activity-aware mental stress detection using physiological sensors”, *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST*, vol. 76 LNICST, pp. 282–301, 2012, doi: 10.1007/978-3-642-29336-8_16/COVER.

[83] T. Penzel, J. W. Kantelhardt, C.-C. Lo, K. Voigt, and C. Vogelmeier, “Dynamics of Heart Rate and Sleep Stages in Normals and Patients with Sleep Apnea”, *Neuropsychopharmacology*, vol. 28, pp. 48–53, 2003, doi: 10.1038/sj.npp.1300146.

[84] D. Žemaityt, G. Varoneckas, K. Plauška, and J. Kauknas, “Components of the heart rhythm power spectrum in wakefulness and individual sleep stages”, *Int. J. Psychophysiol.*, vol. 4, no. 2, pp. 129–141, 1986, doi: 10.1016/0167-8760(86)90006-1.

[85] A. Golgouneh and B. Tarvirdizadeh, “Fabrication of a portable device for stress monitoring using wearable sensors and soft computing algorithms”, *Neural Comput. Appl.*, vol. 32, no. 11, pp. 7515–7537, Jun. 2020, doi: 10.1007/S00521-019-04278-7/FIGURES/23.

[86] R. R. Singh, S. Conjeti, and R. Banerjee, “A comparative evaluation of neural network classifiers for stress level analysis of automotive drivers using physiological signals”, *Biomed. Signal Process. Control*, vol. 8, no. 6, pp. 740–754, Nov. 2013, doi: 10.1016/J.BSPC.2013.06.014.

[87] A. Muaremi, B. Arnrich, and G. Tröster, “Towards Measuring Stress with Smartphones and Wearable Devices During Workday and Sleep”, *Bionanoscience*, vol. 3, no. 2, pp. 172–183, Jun. 2013, doi: 10.1007/S12668-013-0089-2.

[88] “Heart rate variability: standards of measurement, physiological interpretation and clinical use. Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology.”, *Circulation*, vol. 93, no. 5, pp. 1043–65, Mar. 1996, [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/8598068

[89] W. Chen, Z. Wang, H. Xie, and W. Yu, “Characterization of surface EMG signal based on fuzzy entropy”, *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 15, no. 2, pp. 266–272, Jun. 2007, doi: 10.1109/TNSRE.2007.897025.

[90] M. Rostaghi and H. Azami, “Dispersion Entropy: A Measure for Time-Series Analysis”, *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 610–614, May 2016, doi: 10.1109/LSP.2016.2542881.

[91] S. M. Pincus, “Approximate entropy as a measure of system complexity”, *Proc. Natl. Acad. Sci. U. S. A.*, vol. 88, no. 6, pp. 2297–2301, 1991, doi: 10.1073/PNAS.88.6.2297.

[92] J. S. Richman and J. R. Moorman, “Physiological time-series analysis using approximate entropy and sample entropy”, *Am. J. Physiol. Heart Circ. Physiol.*, vol. 278, no. 6, 2000,

doi: 10.1152/AJPHEART.2000.278.6.H2039.

[93] X. Wu *et al.*, “Top 10 algorithms in data mining”, *Knowl. Inf. Syst. 2007 141*, vol. 14, no. 1, pp. 1–37, Dec. 2007, doi: 10.1007/S10115-007-0114-2.

[94] A. Cutler, D. R. Cutler, and J. R. Stevens, “Random Forests”, *Ensemble Mach. Learn.*, pp. 157–175, 2012, doi: 10.1007/978-1-4419-9326-7_5.

[95] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, “A comparative analysis of gradient boosting algorithms”, *Artif. Intell. Rev. 2020 543*, vol. 54, no. 3, pp. 1937–1967, Aug. 2020, doi: 10.1007/S10462-020-09896-5.

[96] Herman *et al.*, “Comparison of Artificial Neural Network and Gaussian Naïve Bayes in Recognition of Hand-Writing Number”, *Proc. - 2nd East Indones. Conf. Comput. Inf. Technol. Internet Things Ind. EIConCIT 2018*, pp. 276–279, Nov. 2018, doi: 10.1109/EICONCIT.2018.8878651.

[97] E. Y. Boateng, J. Otoo, D. A. Abaye, E. Y. Boateng, J. Otoo, and D. A. Abaye, “Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review”, *J. Data Anal. Inf. Process.*, vol. 8, no. 4, pp. 341–357, Sep. 2020, doi: 10.4236/JDAIP.2020.84020.

[98] K. B. Duan and S. S. Keerthi, “Which is the best multiclass SVM method? An empirical study”, *Lect. Notes Comput. Sci.*, vol. 3541, pp. 278–285, 2005, doi: 10.1007/11494683_28/COVER.

[99] M. Feurer and F. Hutter, “Hyperparameter Optimization”, *Automated Machine Learning: Methods, Systems, Challenges*, pp. 3–33, 2019, doi: 10.1007/978-3-030-05318-5_1.

[100] Cburnett, “File:Artificial neural network.svg - Wikimedia Commons”, Dec. 27, 2006. https://commons.wikimedia.org/wiki/File:Artificial_neural_network.svg (accessed Aug. 20, 2022).

[101] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. E. Mohamed, and H. Arshad, “State-of-the-art in artificial neural network applications: A survey”, *Heliyon*, vol. 4, no. 11, p. e00938, Nov. 2018, doi: 10.1016/J.HELIYON.2018.E00938.

[102] A. K. Jain, P. Rao, and K. V. Sharma, “Optimizers in Deep Learning: An Imperative Study and Analysis”, 2021, doi: 10.6025/stm/2021/3/99-106.

[103] P. Baldi and P. Sadowski, “The Dropout Learning Algorithm”, *Artif. Intell.*, vol. 210, no. 1, pp. 78–122, 2014, doi: 10.1016/J.ARTINT.2014.02.004.

[104] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[105] R. Muthukrishnan and R. Rohini, “LASSO: A feature selection technique in predictive modeling for machine learning”, *2016 IEEE Int. Conf. Adv. Comput. Appl. ICACA 2016*, pp. 18–20, Mar. 2017, doi: 10.1109/ICACA.2016.7887916.

[106] T. Hastie, “Ridge Regularization: An Essential Concept in Data Science”, *https://doi.org/10.1080/00401706.2020.1791959*, vol. 62, no. 4, pp. 426–433, Oct. 2020, doi: 10.1080/00401706.2020.1791959.

[107] K. Janocha and W. M. Czarnecki, “On Loss Functions for Deep Neural Networks in Classification”, *Schedae Informaticae*, vol. 25, pp. 49–59, Feb. 2017, doi: 10.48550/arxiv.1702.05659.

[108] S. Raschka and V. Mirjalili, “Python machine learning : machine learning and deep learning with python, scikit-learn, and tensorflow 2”, p. 741, Packt Publishing, 2019, ISBN: 1-78995-829-6.

[109] M. L. McHugh, “Interrater reliability: the kappa statistic”, *Biochem. Medica*, vol. 22, no. 3, p. 276, 2012, doi: 10.11613/bm.2012.031.

[110] S. Sun, “Meta-analysis of Cohen’s kappa”, *Heal. Serv. Outcomes Res. Methodol. 2011 113*, vol. 11, no. 3, pp. 145–163, Nov. 2011, doi: 10.1007/S10742-011-0077-3.

[111] “PLUX Biosignals | Professional Kit.” https://www.pluxbiosignals.com/collections/biosignalsplux/products/professional-kit (accessed Aug. 05, 2022).

[112] “PLUX Biosignals | About Us.” https://www.pluxbiosignals.com/pages/about (accessed Jul. 24, 2022).

[113] “- Sleep as Android.” https://sleep.urbandroid.org/ (accessed Aug. 05, 2022).

[114] B. M. Chaudhry, “Sleeping with an Android”, *mHealth*, vol. 3, pp. 7–7, Feb. 2017, doi: 10.21037/MHEALTH.2017.02.04.

[115] “OpenSignals (r)evolution (Download) – Support PLUX Biosignals official”, Jul. 26, 2022. https://support.pluxbiosignals.com/knowledge-base/introducing-opensignals-revolution/ (accessed Aug. 22, 2022).

[116] “Google Fit | Google Developers.” https://developers.google.com/fit (accessed Aug. 22, 2022).

[117] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python”, *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Jan. 2012, doi: 10.48550/arxiv.1201.0490.

[118] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015, doi: https://doi.org/10.1016/j.neunet.2014.09.003.

[119] H. Sun *et al.*, “Sleep staging from electrocardiography and respiration with deep learning”, *Sleep*, vol. 43, no. 7, 2020, doi: 10.1093/SLEEP/ZSZ306.

[120] N. Sridhar *et al.*, “Deep learning for automated sleep staging using instantaneous heart rate”, *npj Digit. Med.*, vol. 3, no. 1, 2020, doi: 10.1038/s41746-020-0291-x.

[121] S. Basodi, C. Ji, H. Zhang, and Y. Pan, “Gradient amplification: An efficient way to train deep neural networks”, *Big Data Min. Anal.*, vol. 3, no. 3, pp. 196–207, Sep. 2020, doi: 10.26599/BDMA.2020.9020004.

[122] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, “How Does Batch Normalization Help Optimization?”, *Adv. Neural Inf. Process. Syst.*, vol. 2018-December, pp. 2483–2493, May 2018, doi: 10.48550/arxiv.1805.11604.

[123] P. Baldi and P. J. Sadowski, “Understanding Dropout,” in *Advances in Neural Information Processing Systems*, 2013, vol. 26. [Online]. Available: https://proceedings.neurips.cc/paper/2013/file/71f6278d140af599e06ad9bf1ba03cb0-Paper.pdf

[124] O. Walch, Y. Huang, D. Forger, and C. Goldstein, “Sleep stage prediction with raw acceleration and photoplethysmography heart rate data derived from a consumer wearable device”, *Sleep*, vol. 42, no. 12, Dec. 2019, doi: 10.1093/SLEEP/ZSZ180.

[125] S. Kudo, Z. Chen, N. Ono, M. D. Altaf-Ul-Amin, S. Kanaya, and M. Huang, “Deep Learning-Based Sleep Staging with Acceleration and Heart Rate Data of a Consumer Wearable Device”, *LifeTech 2022 - 2022 IEEE 4th Glob. Conf. Life Sci. Technol.*, pp. 305–

307, 2022, doi: 10.1109/LIFETECH53646.2022.9754876.

[126] D. Pollreisz and N. TaheriNejad, “Detection and Removal of Motion Artifacts in PPG Signals”, *Mob. Networks Appl.*, vol. 27, no. 2, pp. 728–738, Apr. 2022, doi: 10.1007/S11036-019-01323-6/TABLES/1.

[127] X. Zhao and G. Sun, “A Multi-Class Automatic Sleep Staging Method Based on Photoplethysmography Signals”, *Entropy*, vol. 23, no. 1, pp. 1–12, Jan. 2021, doi: 10.3390/E23010116.

[128] A. A. Ong and M. B. Gillespie, “Overview of smartphone applications for sleep analysis”, *World J. Otorhinolaryngol. Neck Surg.*, vol. 2, no. 1, pp. 45–49, Mar. 2016, doi: 10.1016/J.WJORL.2016.02.001.

[129] “Sleep as Android: Smart alarm - Apps on Google Play.” https://play.google.com/store/apps/details?id=com.urbandroid.sleep&hl=en_US&gl=US (accessed Aug. 25, 2022).

[130] “Sleep as Android: Smart alarm - Overview - Google Play Store - US.” https://app.sensortower.com/android/US/urbandroid-petr-nlevka/app/sleep-as-android-smart-alarm/com.urbandroid.sleep/overview?tab=about (accessed Aug. 25, 2022).

[131] “Sleep as Android Unlock – Apps no Google Play.” https://play.google.com/store/apps/details?id=com.urbandroid.sleep.full.key&hl=pt&gl=US (accessed Aug. 25, 2022).

[132] “Sleep as Android Unlock - Overview - Google Play Store - US.” https://app.sensortower.com/android/US/urbandroid-petr-nlevka/app/sleep-as-android-unlock/com.urbandroid.sleep.full.key/overview?tab=about (accessed Aug. 25, 2022).

[133] H. M. Lehrer *et al.*, “Comparing polysomnography, actigraphy, and sleep diary in the home environment: The Study of Women’s Health Across the Nation (SWAN) Sleep Study”, *SLEEP Adv.*, vol. 3, no. 1, Jan. 2022, doi: 10.1093/SLEEPADVANCES/ZPAC001.

[134] T. Blackwell *et al.*, “Comparison of Sleep Parameters from Actigraphy and Polysomnography in Older Women: The SOF Study”, *Sleep*, vol. 31, no. 2, p. 283, Feb. 2008, doi: 10.1093/SLEEP/31.2.283.

[135] “How does Sleep as Android compare to the Sleep lab - Sleep as Android.” https://sleep.urbandroid.org/sleep-lab-comparison/ (accessed Sep. 02, 2022).

[136] C. Zhang and Y. Ma, *Ensemble Machine Learning: Methods and Applications*. Springer Publishing Company, Incorporated, 2012, ISBN: 978-1-4419-9325-0.

Appendix A

Table A1- Complementary information about the model's performance briefly described in Figure 4.9. (Bold and red colour for emphasis of model structures with low standard deviation)

Neurons per layer	Max. Accuracy	Avg. Accuracy	Acc. Std. Deviation	Min. Loss	Avg. Loss	Loss Std. Deviation	Max. Cohen's kappa	Avg. Cohen's kappa	Cohen Std. Deviation
16	0,8047	0,6512	0,2095	0,6380	0,9634	0,4325	0,7161	0,4361	0,3744
17	0,8062	0,7260	0,1746	0,6287	0,8057	0,3720	0,7107	0,5667	0,3152
18	0,7947	0,5266	0,1541	0,6653	1,1978	0,3101	0,6934	0,2218	0,2723
19	0,8025	0,7197	0,1723	0,6298	0,8198	0,3674	0,7032	0,5560	0,3109
20	**0,8068**	**0,8026**	**0,0042**	**0,6250**	**0,6414**	**0,0168**	**0,7110**	**0,7035**	**0,0087**
21	0,8041	0,7165	0,1706	0,6309	0,8260	0,3480	0,7064	0,5539	0,3097
22	0,8008	0,6481	0,2078	0,6443	0,9664	0,4342	0,7023	0,4294	0,3704
23	0,8067	0,7261	0,1727	0,6281	0,7988	0,3574	0,7127	0,5684	0,3126
24	**0,8086**	**0,8029**	**0,0042**	**0,6261**	**0,6505**	**0,0176**	**0,7140**	**0,7051**	**0,0050**
25	0,8042	0,6456	0,2130	0,6318	0,9729	0,4483	0,7092	0,4308	0,3762
26	0,8031	0,7243	0,1673	0,6323	0,8024	0,3437	0,7066	0,5676	0,2977
27	0,8013	0,7218	0,1734	0,6456	0,8155	0,3688	0,7089	0,5596	0,3127
28	**0,8039**	**0,8010**	**0,0019**	**0,6448**	**0,6479**	**0,0031**	**0,7074**	**0,7015**	**0,0047**
29	**0,8077**	**0,7993**	**0,0082**	**0,6343**	**0,6568**	**0,0280**	**0,7139**	**0,6995**	**0,0128**
30	0,8054	0,6483	0,2115	0,6272	0,9665	0,4440	0,7098	0,4298	0,3788
31	0,8024	0,7236	0,1679	0,6387	0,8120	0,3512	0,7047	0,5647	0,2977
32	**0,8068**	**0,8023**	**0,0036**	**0,6240**	**0,6480**	**0,0192**	**0,7147**	**0,7059**	**0,0072**
33	0,8054	0,7249	0,1751	0,6325	0,8056	0,3714	0,7123	0,5648	0,3138
34	0,8039	0,7314	0,1501	0,6380	0,8013	0,3231	0,7106	0,5830	0,2616
35	**0,8034**	**0,8014**	**0,0016**	**0,6326**	**0,6408**	**0,0061**	**0,7092**	**0,7039**	**0,0053**
36	**0,8064**	**0,8036**	**0,0027**	**0,6267**	**0,6417**	**0,0144**	**0,7154**	**0,7072**	**0,0054**
37	**0,8027**	**0,8006**	**0,0027**	**0,6390**	**0,6491**	**0,0079**	**0,7117**	**0,7032**	**0,0057**
38	0,8011	0,6586	0,1941	0,6425	0,9493	0,4162	0,7082	0,4523	0,3464
39	0,8026	0,6523	0,2040	0,6441	0,9617	0,4344	0,7017	0,4344	0,3663
40	**0,8058**	**0,8013**	**0,0028**	**0,6314**	**0,6485**	**0,0152**	**0,7102**	**0,7016**	**0,0056**
41	0,7996	0,7212	0,1731	0,6442	0,8186	0,3644	0,7056	0,5592	0,3127
42	0,8045	0,7255	0,1729	0,6307	0,8049	0,3712	0,7084	0,5670	0,3081
43	**0,8045**	**0,8017**	**0,0038**	**0,6300**	**0,6465**	**0,0177**	**0,7116**	**0,6996**	**0,0116**
44	0,8023	0,6552	0,2010	0,6406	0,9583	0,4306	0,7029	0,4427	0,3561
45	0,8021	0,5815	0,1993	0,6423	1,1201	0,4242	0,7025	0,3113	0,3517
46	0,8051	0,6527	0,2025	0,6215	0,9508	0,4248	0,7088	0,4567	0,3426
47	0,8054	0,7244	0,1746	0,6377	0,8094	0,3694	0,7113	0,5656	0,3153
48	0,8033	0,5887	0,1969	0,6444	1,1108	0,4204	0,6998	0,3334	0,3419
49	**0,8068**	**0,8024**	**0,0029**	**0,6306**	**0,6405**	**0,0061**	**0,7122**	**0,7019**	**0,0066**
50	**0,8034**	**0,8011**	**0,0029**	**0,6375**	**0,6450**	**0,0083**	**0,7078**	**0,7012**	**0,0066**
51	**0,8046**	**0,8001**	**0,0047**	**0,6368**	**0,6526**	**0,0221**	**0,7053**	**0,6990**	**0,0063**
52	0,8031	0,7218	0,1728	0,6429	0,8151	0,3617	0,7049	0,5603	0,3100
53	0,8051	0,7425	0,1347	0,6285	0,7604	0,2709	0,7120	0,6095	0,2202
54	0,8014	0,7323	0,1516	0,6326	0,7978	0,3233	0,7091	0,5988	0,2310
55	**0,8075**	**0,8015**	**0,0048**	**0,6329**	**0,6466**	**0,0140**	**0,7093**	**0,7014**	**0,0064**
56	**0,8040**	**0,8012**	**0,0021**	**0,6375**	**0,6413**	**0,0065**	**0,7099**	**0,7020**	**0,0060**
57	**0,8031**	**0,8016**	**0,0015**	**0,6434**	**0,6480**	**0,0043**	**0,7119**	**0,7039**	**0,0059**
58	**0,8064**	**0,7986**	**0,0073**	**0,6337**	**0,6632**	**0,0240**	**0,7098**	**0,6974**	**0,0104**
59	**0,8060**	**0,8011**	**0,0033**	**0,6359**	**0,6432**	**0,0084**	**0,7089**	**0,7051**	**0,0024**
60	0,8065	0,7302	0,1609	0,6316	0,7917	0,3400	0,7101	0,5735	0,2868
61	0,8014	0,6548	0,1999	0,6365	0,9580	0,4270	0,7036	0,4396	0,3592
62	0,8039	0,7267	0,1708	0,6348	0,8023	0,3595	0,7087	0,5705	0,3033
63	**0,8050**	**0,8013**	**0,0030**	**0,6267**	**0,6428**	**0,0139**	**0,7124**	**0,7067**	**0,0070**
64	**0,8057**	**0,8026**	**0,0026**	**0,6354**	**0,6435**	**0,0066**	**0,7114**	**0,7055**	**0,0040**

Table A2- Complementary information about the model's performance briefly described in Figure 4.10. (Bold and red colour for models with maximum accuracy above 70%)

Dropout Value	Max. Accuracy	Avg. Accuracy	Acc. Std. Deviation	Min. Loss	Avg. Loss	Loss Std. Deviation	Max. Cohen's	Avg. Cohen's kappa	Cohen Std. Deviation
0.00	**0.7890**	**0.7037**	**0.1702**	**0.6368**	**0.8078**	**0.3346**	**0.7141**	**0.5712**	**0.2904**
0.05	**0.7655**	**0.4184**	**0.1233**	**0.7071**	**1.3573**	**0.2322**	**0.6827**	**0.0849**	**0.2161**
0.10	**0.7427**	**0.4103**	**0.1168**	**0.7554**	**1.3742**	**0.2174**	**0.6356**	**0.0636**	**0.2010**
0.15	**0.7556**	**0.4117**	**0.1208**	**0.7323**	**1.3735**	**0.2253**	**0.6375**	**0.0641**	**0.2015**
0.20	**0.7039**	**0.4065**	**0.1045**	**0.8759**	**1.3878**	**0.1799**	**0.5566**	**0.0560**	**0.1759**
0.25	0.3735	0.3734	0.0001	1.4438	1.4460	0.0013	0.0003	0.0001	0.0001
0.30	0.3737	0.3734	0.0001	1.4434	1.4457	0.0013	0.0009	0.0002	0.0003
0.35	0.3739	0.3734	0.0002	1.4439	1.4461	0.0015	0.0005	0.0002	0.0002
0.40	0.3735	0.3733	0.0001	1.4395	1.4449	0.0030	0.0002	0.0001	0.0001
0.45	0.3734	0.3733	0.0000	1.4443	1.4461	0.0012	0.0002	0.0000	0.0001
0.50	0.3736	0.3733	0.0001	1.4423	1.4459	0.0019	0.0003	0.0001	0.0001

Table A3- Complementary information about the model's performance briefly described in Figure 4.13.

L1 Value	Max. Accuracy	Avg. Accurac	Acc. Std. Deviation	Min. Loss	Avg. Loss	Loss Std. Deviation	Max. Cohen's kappa	Avg. Cohen's	Cohen Std. Deviation
0.00	0.7929	0.7262	0.1135	0.6372	0.8096	0.2872	0.7154	0.5523	0.2671
0.05	0.7970	0.7182	0.1263	0.7451	0.9547	0.2747	0.7175	0.5361	0.2923
0.10	0.7943	0.6766	0.1437	0.7978	1.0771	0.2959	0.7137	0.4432	0.3311
0.15	0.7934	0.7059	0.1281	0.8382	1.0533	0.2604	0.7144	0.5109	0.2935
0.20	0.7922	0.6153	0.1450	0.8968	1.2496	0.2728	0.7085	0.3004	0.3328
0.25	0.7928	0.6076	0.1363	0.9420	1.3099	0.2409	0.7097	0.2796	0.3107
0.30	0.7882	0.5673	0.1177	0.9898	1.3975	0.2044	0.7056	0.1938	0.2650
0.35	0.7608	0.5385	0.0817	1.1244	1.4880	0.1157	0.6090	0.1415	0.1885
0.40	0.7918	0.5365	0.0819	1.1160	1.5127	0.1031	0.7048	0.1260	0.1916
0.45	0.7605	0.5069	0.0427	1.2053	1.5503	0.0500	0.6056	0.0650	0.1098
0.50	0.5955	0.5011	0.0257	1.5616	1.5718	0.0040	0.3906	0.0492	0.0914

Table A4- Complementary information about the model's performance briefly described in Figure 4.14.

L2 Value	Max. Accuracy	Avg. Accurac	Acc. Std. Deviation	Min. Loss	Avg. Loss	Loss Std. Deviation	Max. Cohen's	Avg. Cohen's kappa	Cohen Std. Deviation
0.0	0.7935	0.7890	0.0071	0.6415	0.6403	0.0140	0.7048	0.6969	0.0149
0.1	0.7973	0.7925	0.0023	0.6609	0.6709	0.0051	0.7212	0.7181	0.0016
0.2	0.7985	0.7953	0.0027	0.6642	0.6751	0.0083	0.7241	0.7206	0.0028
0.3	0.7988	0.7941	0.0050	0.6629	0.6795	0.0155	0.7226	0.7187	0.0046
0.4	0.8006	0.7946	0.0041	0.6585	0.6889	0.0178	0.7246	0.7187	0.0044
0.5	0.7967	0.7932	0.0032	0.6645	0.6951	0.0170	0.7219	0.7171	0.0042

Table A5- Complementary information about the model's performance briefly described in Figure 4.15. (Bold and red colour for emphasis of the value where the drop in performance begins)

L2 Value	Max. Accuracy	Avg. Accurac	Acc. Std. Deviation	Min. Loss	Avg. Loss	Loss Std. Deviation	Max. Cohen's	Avg. Cohen's kappa	Cohen Std. Deviation
0.0	0.8085	0.8059	0.0025	0.6149	0.6284	0.0101	0.7176	0.7106	0.0069
0.1	0.8122	0.8093	0.0016	0.6426	0.6573	0.0109	0.7195	0.7160	0.0028
0.5	0.8114	0.8104	0.0007	0.6750	0.6855	0.0068	0.7195	0.7168	0.0019
1.0	0.8114	0.8038	0.0184	0.6865	0.7300	0.0541	0.7198	0.7051	0.0330
2.0	**0.8104**	**0.5832**	**0.1723**	**0.7482**	**1.2119**	**0.3376**	**0.7154**	**0.3218**	**0.3006**
3.0	0.4746	0.4204	0.0200	1.4704	1.4736	0.0018	0.1585	0.0318	0.0509
4.0	0.4714	0.4213	0.0197	1.4698	1.4727	0.0016	0.1143	0.0213	0.0422
5.0	0.5084	0.4357	0.0309	1.4694	1.4720	0.0018	0.1786	0.0693	0.0591
10.0	0.4415	0.4209	0.0121	1.4713	1.4737	0.0012	0.1001	0.0235	0.0332
50.0	0.4250	0.4134	0.0044	1.4705	1.4735	0.0015	0.0525	0.0099	0.0210