Minor Effects of COVID-19 on Objective Sleep Parameters: Indications for Changes in EEG Morphology

by

Katrín Hera Gústafsdóttir

Thesis of 30 ECTS credits submitted to the School of Science and Engineering
at Reykjavík University in partial fulfillment
of the requirements for the degree of
Master of Science (M.Sc.) in Mechatronics Engineering

June 2022

Supervisor:

Erna Sif Arnardóttir, Supervisor
Assistant Professor, Reykjavík University, Iceland

Karl Ægir Karlsson, Supervisor
Professor, Reykjavík University, Iceland

María Óskarsdóttir, Supervisor
Assistant Professor, Reykjavík University, Iceland

Examiner:

Timo Leppänen, Examiner
Associate Professor, University of Eastern Finland, Kuopio, Finland

Copyright
Katrín Hera Gústafsdóttir
June 2022

Minor Effects of COVID-19 on Objective Sleep Parameters: Indications for Changes in EEG Morphology

Katrín Hera Gústafsdóttir

June 2022

Abstract

It has been suggested that COVID-19 has a negative effect on sleep quality. Whether it is caused by COVID-19 infection or other pandemic related factors remains to be determined. Data from DeCODE's Icelandic Health Study offers an opportunity for such analysis where self applied sleep studies were conducted on 33 participants on two separate occasions and of those 14 contracted COVID-19 in between the occasions.

The objectives were twofold, firstly to run traditional statistical analyses on sleep quality variables. The variables of interest included measures of total sleep time, sleep efficiency, rapid eye movement (REM) and deep sleep, arousal index, sleep latency and sleep apnea severity. Both parametric and non-parametric analyses were conducted. The secondary objectives were to extract features from 30 second signal epochs and train machine learning models to classify whether the features are from studies before or after COVID-19.

The results of the paired t-tests, indicated significant difference in one variable: elevated mean saturation. However, a power analysis showed that the paired t-tests had insufficient power to detect the changes if present. The non-parametric test, aligned rank transform (ART) followed by analysis of variance (ANOVA), found no significant differences for the two studies in terms of any of the variables of interest. Therefore, it cannot be stated with any confidence that COVID-19 has negative effects on sleep quality from the tests performed.

Three types of machine learning models, Decision Trees, Random Forest and Multilayer Perceptron, were trained on six different datasets depending on the sleep stage. The datasets consisted of five datasets with epochs from different sleep stages or wake (N1, N2, N3, REM and wake) and the whole dataset consisting of all epochs. The performance of the Random Forest classifier was consistently best for all datasets. The highest performance was achieved with the deep sleep (N3) dataset with performance metrics high enough to be consistently better than random guessing. These results might indicate changes in the morphology of EEG signals between the two studies especially when trained on epochs from deep sleep and should therefore warrant further investigation.

Minniháttar áhrif COVID-19 á mældan svefn: Vísbendingar um breytingar á formfræði heilarafrits

Katrín Hera Gústafsdóttir

júní 2022

Útdráttur

Vísbendingar eru um að COVID-19 sýking hafi neikvæð áhrif á gæði svefns. Hvort þessi neikvæðu áhrif séu vegna COVID-19 sýkingar eða annarra COVID-19 faraldurs tengdra þátta er enn óljóst. Gögn frá Heilsurannsókn Íslenskrar Erfðagreiningar bjóða uppá tækifæri til að skoða þessi tengls nánar. Framkvæmdar voru sjálfuppsettar svefnmælingar á 33 einstaklingum í tvígang. Af þessum 33 einstaklingum fengu 14 af þeim COVID-19 á milli svefnmælinga.

Markmið þessa verkefnis eru tvíþætt. Annars vegar að kanna með tölfræðiprófum hvort COVID-19 hafi áhrif á svefngæði með tilliti til svefn lengdar, skilvirkni svefns, draumsvefns (e. REM) og djúpsvefns, uppvaknana, erfiðleika við að festa svefn, súrefnismettunar- og hrotumælinga. Framkvæmd voru bæði stika- og stikalaus próf. Hinsvegar er markmiðið að nota mismunandi tegundir af vitvélum til að greina 30 sekúnda búta(e. epochs) af heilarafriti (e. electroencephalography). Merkjaeiginleikar heilarafrits voru dregnir út úr hverjum bút og nýttir til að þjálfa vitvélar í að þekkja hvort mælingin hafi verið gerð fyrir eða eftir COVID-19 sýkingu.

Niðurstöður úr pöruðu t-prófi bentu til þess að ein breyta, meðal súrefnismettun, hafi hækkað milli mælinga með marktækum mun. Próf til að kanna tölfræðilegan styrk paraða t-prófsins var framkvæmt vegna smæðar gagnanna og kom þar í ljós vöntun á tölfræðilegum styrk til að merkja mun ef hann er til staðar. Stikalausa prófið sem framkvæmt var til viðbótar kallast fervikagreining á gögnum sem búið var að stilla með tilliti til háðu þáttanna(e. ART ANOVA). Niðurstöður úr fervikagreinungunum sýndu engan marktækan mun á neinum af breytunum sem prófaðar voru. Því var ekki hægt að draga neinar ályktanir um áhrif COVID-19 á svefngæði út frá þessum niðurstöðum.

Þrjár mismunandi gerðir af vitvélum voru þjálfaðar, ákvörðunartré (e. Decision Tree), slembiskógur (e. Random Forest) og marglaga tauganet(e. Multilayer Perceptron), á sex mismunandi gagnasettum. Gagnasettin samanstóðu af eiginleikum heilarafritsbúta úr fimm mismunandi svefnstigum (N1, N2, N3, REM og vaka) og einnig öllum bútum úr öllum svefnstigum saman. Vitvélin slembiskógur sýndi besta árangurinn með tilliti til allra gangasetta. Niðurstöðurnar bentu til þess að þjálfun á djúpsvefnsgagnasettinu (N3) hlaust besti árangurinn en árangurinn var nógu góður til að teljast betri en flokkun af handahófi. Það er því ljóst að vert er að gera frekari athugun á heilarafristgögnum í sambandi við breytingar á heilarafriti svefns eftir COVID-19 sýkingu.

Acknowledgements

This work was financially supported by the European Union's Horizon 2020 Research and Innovation Programme (grant 965417) and the Nordforsk (NordSleep project 90458) via the Icelandic Research Fund.

I would like to thank my supervisors Erna Sif Arnardóttir, María Óskarsdóttir and Karl Ægir Karlsson for their support and guidance. Special thanks to the expert sleep technologists from the Sleep Revolution who scored the studies, Sigríður Sigurðardóttir, Heiður Grétarsdóttir, Kristín Anna Ólafsdóttir and Tiina Siilak, without you this work would not have been possible. Further I would like to thank Kári Stefánsson and Hilma Hólm for trusting me with their data and allowing me access to their facilities.

Finally, I would like to thank my nearest and dearest for their support and encouragement. You know who you are.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

There are currently six known types of human transmissible coronaviruses, of which two types cause severe respiratory disease: SARS-CoV and MERS-CoV [1], [2]. The novel coronavirus responsible for the 2020 COVID-19 pandemic, SARS-CoV-2, in its acute stage causes infiltrated pneumonia, gastrointestinal symptoms, and in some cases neurological complications. Coronaviruses have been known to cause neurological symptoms and that includes SARS-CoV-2, the most common neurological symptom being loss of smell [3]. SARS-CoV-2 affects the respiratory and gastrointestinal tracts and evidence suggests that it also invades the central nervous system (CNS) of the host. It has been suggested that the coronaviruses' entry into the CNS is by olfactory transmucosal invasion using the Angiotensin converting enzyme 2 (ACE2) receptor and therefore bypassing the blood-brain-barrier [3], [4]. The ACE2 receptor is expressed in the respiratory and intestinal tracts, heart, kidneys, and brain of humans which makes those systems consequently vulnerable to SARS-CoV-2 infection [5]. Respiratory failure in COVID-19 patients may not be exclusively due to pneumonia as coronaviruses have been shown to infect the medullary cardiorespiratory center of the brain and thus, causing respiratory failure [6], [7].

Two main routes to neuronal damage are proposed following SARS-CoV-2 infection: hypoxic brain injury and immune-mediated damage. In terms of hypoxic damage, peripheral vasodilatation, hypercapnia, hypoxia, and anaerobic metabolism have been postulated as the main mechanisms. The main mechanisms in terms of immune-mediated damage, vastly increased release of inflammatory cytokines and activation of T lymphocytes, macrophages, and endothelial cells have been proposed [8]. Neurological symptoms include: dizziness, headache, acute cerebrovascular disease, impaired consciousness, psychiatric episodes, lethargy, transverse myelitis, acute hemorrhagic necrotizing encephalopathy, encephalopathy, encephalitis, epilepsy, ataxia, hypogeusia, hyposmia, neuralgia and Guillian-Barré syndrome [9]–[14]. Thirty-seven severely ill COVID-19 patients with neurological complications, excluding ischemic infarcts, were found to have abnormal brain MRI scans with distinctive lesion patterns [15]. Furthermore, hospitalized COVID-19 patients with neurological complications were shown to have worse functional outcomes than those without neurological complications. The COVID-19 patients with neurological complications had abnormal functional outcomes 6 months after hospitalization including depression, anxiety, and sleep difficulties and 47% of the subjects could not return to work 6 months after discharge from hospital [16]. It is therefore reasonable to assume that the long-term symptoms of COVID-19 may at least in part be due to irreversible neuronal damage, particularly for those needing intensive care.

There has been growing concern regarding the persistent symptoms of COVID-19, in particular the neurological symptoms [17]. The prevalence has not been fully determined, but

studies suggest a prevalence ranging from 72% to 80% of COVID-19 survivors having at least one persistent long-term symptom [18]–[20]. The most common long-term symptoms were fatigue, "brain fog", dizziness, headache, attention disorder, hair loss and dyspnea [19], [21]. However, it is likely that the well-known post intensive-care syndrome accounts for some of the long-term deleterious impairments suffered by hospitalized COVID-19 patients who received mechanical ventilation due to respiratory failure [22]. Yet, not all who suffer from long-term COVID-19 symptoms have received critical care and therefore, it can be argued that post intensive-care syndrome is not the only explanation for the long-term symptoms of COVID-19. A study found that there was little difference in long-term symptoms of non-hospitalized COVID-19 patients compared to hospitalized patients, and all scored worse on cognitive and quality of life assessments compared to a demographically matched reference population [21].

The physiology of sleep is complex and is regulated by multiple neuronal tracts in the brain. To initiate sleep the so-called ascending arousal system, an extensive network of subcortical structures, needs to be inhibited. The inhibitory neurons of the ventrolateral preoptic area of the brain hinder the activation of the ascending arousal system, thus initiating and maintaining sleep [23]. Since COVID-19 is causing neurological complications, it can be reasoned that the higher rates of insomnia and sleep disturbances in COVID-19 survivors could be in part due to the disruption of neural tracts in the brain involved in maintaining sleep.

Studies suggest that COVID-19 has had diverse effects on those it afflicts. Some individuals report no lasting symptoms while others suffer from a multitude of long-term problems. According to a recent meta-analysis, 72% of COVID-19 survivors report having at least one persistent long-term symptom [18]. Among the long-term symptoms reported were insomnia and other sleep disorders. Furthermore, 57% of subjects with long-term COVID-19 symptoms reported having problems with sleep, according to a systematic review [24].

Normal human sleep is commonly divided into two types of sleep: rapid eye movement (REM) sleep and non-REM (NREM) sleep. REM sleep is associated with dreaming while NREM sleep is further subdivided into progressively deepening sleep stages; N1, N2 and N3, where N1 is the lightest sleep stage and N3 is deep sleep often referred to as slow wave sleep. Normal sleep alternates between all sleep stages in cycles of 60-90 minutes throughout the night where N3 and REM are most prominent in the beginning and end of the night, respectively. Therefore, measuring how much time spent in each sleep stage is one of the variables used to determine objective sleep quality [25], [26].

Polysomnography (PSG) is the golden standard for objective sleep measurements, and it involves noninvasive electrodes placed on the subject measuring electroencephalogram (EEG), electrooculogram (EOG), electrocardiogram (ECG), submental and leg electromyogram (EMG) along with respiratory parameters [23]. EEG along with EOG and EMG is used to quantify different sleep stages [27]. Due to technological advances, PSG, previously only conducted in a laboratory environment, is now available for self application and home use [28].

Each sleep stage can be identified by certain characteristics of the EEG and EOG signals. At sleep onset the fast frequency and low-voltage alpha waves of wake give way to slower alpha frequencies accompanied by rolling eye movements as the first and lightest sleep stage, N1, takes over. N2 is characterized by theta waves and phenomena called K-complexes and spindles. The deepest sleep stage, N3, has high amplitude and slow frequency delta waves. REM sleep is so called because of the rapid eye movements and more erratic and fast frequency of the EEG. Other signs to confirm REM sleep is the loss of muscle tone seen in

the EMG signals. The wake scoring refers both to brief arousals and longer periods of wake during the night if present [25].

Sleep plays a vital role in physical and psychiatric health. The three main components of good sleep hygiene are: duration, quality, and timing consistency [29]. Long-term difficulties with sleep have been associated with a multitude of health problems such as: obesity, hypertension, heart disease, stroke, diabetes, depression, anxiety, and increased risk of death [30].

Sleep is a complex physiological function that is highly regulated in the central nervous system on a molecular level. It is one of the universal rules in the animal kingdom, all complex biological beings need sleep to some extent [31]. Lack of sleep has been shown to have negative effects on health and cognitive function and has also been shown to weaken the immune system and elevate the risk of cancer [31], [32].

Environmental factors in a pandemic could be a substantial contributor to higher rates of insomnia and sleeping difficulties. Factors such as lack of direct sunlight due to home confinement and depression caused by lack of social contact and pandemic related stress could contribute to the increase in sleeping difficulties. During the COVID-19 surge in Italy in March 2020, people in home confinement were experiencing poorer sleep quality even though they spent more time in bed than usual [33]. The Insomnia Severity Index was used in China to survey the subjective sleep status in a COVID-19 affected population and 20% were found to have clinical insomnia [34] and over a third of health-care workers in China had insomnia during the COVID-19 outbreak [35]. The pooled prevalence of sleep disturbances in multinational pandemic affected populations was found to be 34% and insomnia 23% according to a meta-analysis [36]. Health-care workers had higher rates of insomnia compared to non-health-care workers [37]. Sleep disturbances have many causes, and it is well documented that traumatic events impact sleeping patterns and can cause sleep disturbances [38]. It can be argued that the COVID-19 pandemic is causing collective trauma to whole societies which could be a possible contributor to the higher rates of insomnia [39].

The effects of viral infections on objective sleep are not well known, but subjective measures from COVID-19 survivors suggest negative impact on sleeping patterns [16], [36]. It is therefore reasonable to assume that COVID-19 infection or COVID-19 related factors have negative effects on subjective sleep but further research is needed to investigate whether a relationship between objective sleep measures and long-term COVID-19 symptoms exists.

Viral infections, in particular influenza and human immunodeficiency virus infections have been shown to change sleep in humans. Increased N3 in the second half of night, increased night-time awakenings and changes in REM sleep patterns are among the effects described [40], [41]. A bidirectional relationship has been proposed between sleep and the immune system. When the immune system is activated by a pathogen it can either cause enhanced sleep or disrupted sleep [42]. The main theory of why sleep is enhanced during an immune response is to promote host defenses but, regarding why sleep disruption occurs, the assumption is that it depends on the severity of the infection and type of pathogen [42]. Moreover, lack of sleep can increase susceptibility to infections due to neuroinflammation and blood-brain-barrier leakage of antigens and inflammatory factors into the brain [43], [44].

Machine learning in medicine has widely been found powerful to identify clinically relevant patterns. Its uses vary from classification of medical images for diagnosing purposes to aggregation of data to predict outcome, behaviour, disease prevention and treatment. Their computational capabilities allow for concise pattern recognition and interpretation on large heterogenous biomedical datasets [45]. Machine learning algorithms have been shown to accurately detect sleep disorders such as obstructive sleep apnea [46] and detect seizure

activity in EEG signals [47], [48]. Automatic sleep staging using various deep learning methods has been well established with promising performance [49]. Since the standard practice is for sleep experts to visually score sleep stages, it could be theorized that the EEG signal, the main signal for sleep staging, may contain additional patterns undetectable by the human eye. Such patterns might allows us to understand the microstructure of sleep further.

Various mathematical methods for extracting information from big datasets, generally termed machine learning, have been described. Machine learning methods are commonly divided into three types: supervised, unsupervised and reinforcement learning. Supervised learning algorithms learn from a set of labelled feature vectors while unsupervised learning algorithms use non-labelled features to discover patterns in the data. Reinforcement learning involves the algorithm learning through trial and error over time using reward maximisation [50]. Two subfields of supervised learning are classification and regression and the main difference between them being that the target variable for classification is discrete while for regression the target is continuous. The general supervised learning procedure is to fit the feature data of the target variable to the model in a process called model training. Afterward, its performance can be evaluated by predicting the target variable using unlabelled data not used in the training. Classification models take in an unlabelled feature vectors and output a discrete prediction of the target variable commonly termed class label. While regression models also take in an unlabelled feature vector the prediction is not a class label but a real valued continuous target variable [50], [51].

Three models are proposed for investigation of changes in EEG morphology: Decision Tree, Random Forest and a neural network called Multilayer Perceptron (MLP). Decision trees are weak learners but yet they are popular decision making tools. Their name is derived from their growing nature. From the root node the input data is split according to a decision criteria at each node. Eventually, the splits stop according to certain stopping parameters [50]. Ensemble models are so called because they are constructed of a collection of weak learners. One such model is the Random Forest which is a tree based classification method where a combination of decision trees are deployed in tandem to form a majority decision classification [52]. Neural networks get their name from their similarity to the human nervous system. The endeavour of finding a mathematical description of the brain's ability to process information was the inception of neural networks in machine learning [50]. Neural networks are essentially models that perform series of functional transformations on an input of independent variables. In the first layer the input variables are transformed into linear combinations termed weights and biases. At each layer in the network the weights and biases are passed through nonlinear activation functions [50].

Error metrics are important for evaluation of machine learning model performance. The most important performance metrics are based on the confusion matrix, which is a matrix depicting how many observations the model has correctly or incorrectly classified. For binary classification problems the F1-score and AUC are the most robust performance metrics because they are less likely to be affected by class imbalance. However, it is not wise to rely too heavily on one performance metric and therefore studying the confusion matrix and the derived metrics mentioned to gain information on the prediction behaviour of the estimator is useful[53], [54].

The unique opportunity has presented itself in the data from DeCODE's Icelandic Health Study, where self administered sleep studies were conducted on individuals before and after contracting COVID-19. This may reveal if long term effects of COVID-19 cause disruption of normal sleep patterns.

The primary objective was to investigate whether there is a detectable difference in sleep quality before and after COVID-19 infection. Both parametric and non-parametric

analyses were conducted to investigate whether a difference is present. Since there is evidence that viral infections may affect certain aspects of sleep are evaluated such as sleep duration, fragmentation and sleep stages. In particular, changes in REM and deep sleep have been linked with viral infections and are thus specially analysed. Additionally, due to the respiratory symptoms caused by COVID-19 the oxygen saturation measures are included in the analysis. The main hypothesis is that COVID-19 causes shorter sleep time, decreased sleep efficiency and poorer sleep quality in terms of REM and deep sleep as well as increased sleep fragmentation (arousals). The hypothesis further postulates that the oxygen saturation measures are lowered and mean heart rate elevated in response to COVID-19.

The secondary objective was to use the aforementioned machine learning methods on EEG signal features to discern possible changes in the EEG signal. Classifiers are trained on EEG signal epoch features from different sleep stages to predict whether features are from a sleep measurement before or after COVID-19 infection. The aim is, therefore, to investigate the possible effects of a COVID-19 infection on sleep quality by detecting patterns in EEG signals that are otherwise undetectable by the human eye. Therefore, the secondary hypothesis is that COVID-19 causes changes in the EEG signal morphology and the microstructure of sleep.

Chapter 2

Methods

2.1 Cohort and Data Collection Methods

The cohort consisted of 33 participants in the Icelandic Health Study designed by DeCODE (Reykjavik, Iceland) that had completed two separate one-night sleep studies in the time period from December 2017 to September 2021. Of the 33 participants 14 were diagnosed with COVID-19 in between the two studies. The Icelandic Health Study is a study designed to understand the genetic risk factors of diseases in the Icelandic population. The study includes several measurements ranging from sight, smell and hearing to a neurological workup, exercise stress tests and sleep studies [55], [56].

The inclusion criteria set for this study was that the PSG needed to be scoreable by a sleep technologists and at least 4 hours of sleep per recording. The initial cohort consisted of 33 participants (n=66) but due to poor EEG signal quality, 10 of the 66 sleep studies were not scoreable and thus excluded from analysis and further 8 studies were excluded because either the first or second study had been previously excluded. The resulting cohort therefore consisted of 24 participants (n=48), of which 11 had COVID-19 between studies. A cohort flowchart can be seen in Figure 2.1. For the secondary objectives, 4 of the 11 measurement pairs that had COVID-19 in between studies, one or both of the raw signal recordings were lost and therefore only 7 of 11 study pairs were used in the machine learning part of the thesis.

2.1.1 Self Applied Somnography

The measurement equipment used was the Self Applied Somnography (SAS) setup from Nox Medical (Nox Medical, Reykjavik, Iceland) which can be seen in Figure 2.2a. As the name implies the equipment is designed to be self applied and worn while sleeping at home. The setup is equivalent to a type II polysomnography setup but with a simplified forehead EEG and the chin EMG omitted. It further includes sound recording and breathing measurements; oxygen saturation and pulse, airflow through the nose and breathing movements via thoracic and abdomen belts. The simplification of the EEG equipment results in a reduction of power in the signal, but has been shown to be adequate for sleep stage classification [57]. The simplified forehead EEG is shown in Figure 2.2b. The forehead EEG includes four EEG channels and four EOG channels.

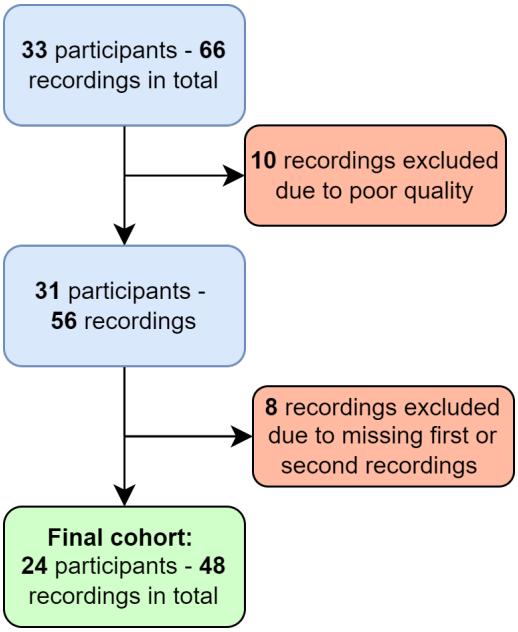
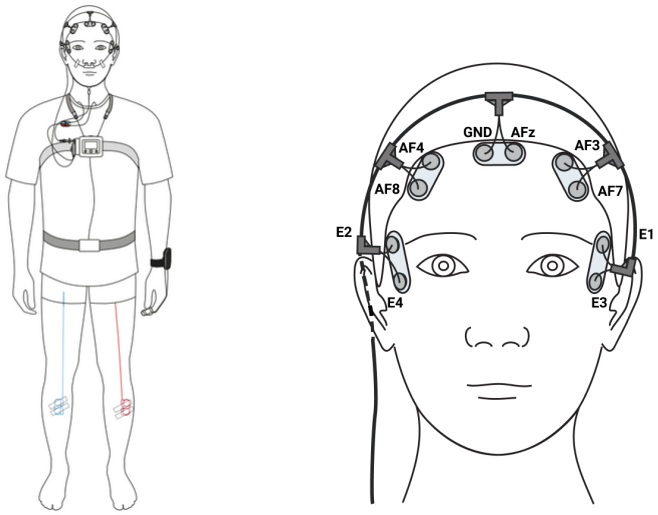


Figure 2.1: Cohort flowchart including the two exclusion criteria.



(a) Self Applied Somnography full setup.

(b) Simplified forehead electroencephalography (EEG) setup. The EEG channels are labelled AF3, AF4, AF7, AF8.

Figure 2.2: Self applied somnography full body setup and electroencephalography configuration. Figures from Nox Medical, Reykjavik, Iceland.

Table 2.1: Overview of primary and secondary evaluation variables.

Primary Variables	Secondary Variables
Total Sleep Time [min]	Sleep Latency [min]
Sleep Efficiency [%]	Arousal Index [arousals/hour]
N3 Stage [min]	AHI
REM Stage [min]	ODI
Mean Oxygen Saturation [%]	Oxygen Saturation below 90% [min]
Mean Pulse [bpm]	

Abbreviations: REM, rapid eye movement; ODI, Oxygen desaturation index; AHI, Apnea-hypopnea index; bpm, beats per minute.

2.2 Sleep Data Annotation

Annotation of the data was performed by expert sleep technologists in accordance with version 2.6 of the American Academy of Sleep Medicine(AASM) scoring manual [58] using Noxturnal software, version 6.2.2 (Nox Medical, Reykjavik, Iceland). In addition to sleep stages, breathing events such as snoring, apneas and hypopneas were scored. After sleep scoring summary values of total sleep time (TST), sleep efficiency (SE) and duration in each of the sleep stages were calculated. Oxygen saturation and pulse measurements were also processed and the average measures of oxygen saturation, time of oxygen saturation below 90% and pulse were automatically calculated over the sleep period. Other derived sleep quality measures were sleep latency(SL) and arousal index (AI). SE is defined as the ratio between total sleep time and time spent in bed and AI is a measure of sleep fragmentation. It is defined as the number of arousals per hour of sleep [59], [60]. Sleep latency refers to the time from trying to fall a sleep until first stage of light sleep is detected [61]. The primary variables for evaluation were chosen to best reflect sleep quality and respiratory health. The variables chosen as primary were TST, SE, duration in stages N3 and REM, mean oxygen saturation and mean pulse. The secondary variables of evaluation were SL, AI, apnea-hypopnea index (AHI), oxygen saturation index (ODI), and duration of oxygen saturation being below 90%. AHI is defined as the number of apneas and hypopneas per hour of sleep [62]. ODI is a measure of the number of oxygen saturation decreases by $\geq 3\%$ per hour [63]. Table 2.1 summarizes the primary and secondary variables of evaluation in the primary objectives of this thesis.

2.3 Statistical Analyses

The significance threshold in the statistical analyses was chosen to be <0.05 because of the small number of study pairs (n=24). All statistical and power analyses in this thesis were conducted in Python 3.9 [64] with the exception of the Alinged Rank Transform ANOVA discussed in section 2.3.2 which was performed in R 3.6.3 [65].

2.3.1 Parametric Analysis

When investigating the possible changes between the sleep studies, a paired t-test was used. Since the number of observations is small for all variables of interest, the likelihood of detecting significant differences between the studies using the paired t-test is low due to deficiency in analysis power [66]. Accordingly, a power analysis was conducted with p<0.05

to evaluate both the power of the paired t-test and to investigate how big the dataset would need to be to get at least 80 % power. The power analysis was performed using the Python package *pingouin* [67].

2.3.2 Aligned Rank Transform Analysis of Variance (ART ANOVA)

As a non-parametric alternative, full factorial analyses of variance (ANOVA) were applied with prior aligned rank transform(ART) of the data to facilitate its multifactorial nature. The ART ANOVAs were done using ARTool package in R [68].

Non-parametric tests such as Mann-Whitney U and Wilcoxon are usually one-way and therefore only take one factor into account per analysis. The aligned rank transform was invented to bridge that gap and enable easier comparison for multifactorial data. The procedure consists of alignment steps followed by a ranking step. The data in this study has two factors, COVID-19 and measurement number. The levels of the COVID-19 factors are before COVID-19 infection and after COVID-19 infection. The measurement factor has the levels first or second measurement. Three ANOVAs are performed for each of the primary and secondary variables. First, the data is aligned and ranked solely in regards to the COVID-19 factor and will only measure the effect of that factor. For the second ANOVA the data is aligned and ranked in regards to the measurement factor and the third ANOVA is with regards to both factors which therefore measures the interaction effect of the two factors.

2.4 Machine Learning Methods

The data in this part of the study consists of the raw EEG signals from seven of the COVID-19 affected subjects, where each subject had two nights worth of EEG signals comprising 4 channels. The data can therefore be split into two classes; No COVID-19 class consisting of seven recordings from before contacting COVID-19 and COVID-19 class which consists of epochs from seven recordings after contracting COVID-19. The machine learning methods used in this thesis are three different supervised classification models: Decision Tree [69], Random Forest [52] and Neural Network called Mulitlayer Perceptron [70].

2.4.1 Signal Preprocessing and Feature Extraction

The raw signals were exported from the Noxturnal software with a sampling frequency of 200 Hz. The signal was trimmed to fit the start and end of annotation from the scoring files and split into 30 second epochs where only scored epochs were included. Each epoch was assigned a sleep stage according to the annotation and a label of either no COVID-19 class or COVID-19 class.

In addition to using the whole dataset, the dataset was also further split into five smaller datasets according to their scored sleep stage (N1, N2, N3, REM and Wake). Therefore, six datasets were trained with the three machine learning methods proposed. Table 2.2 outlines the composition of the six datasets. The binary class distribution of each dataset gives a benchmark for the prediction accuracy, their goal is to predict better than random guessing and therefore the accuracy to beat is the class distribution.

Feature extraction of signals relates to extraction of descriptive information from raw signals where each feature gives insight into a specific aspect of that signal. Features of signals can be subdivided into four categories: time domain features, frequency domain features, time-frequency domain features and nonlinear features [71]. In this thesis, the final

Table 2.2: Total number of epochs and ratio of epochs in class after COVID-19.

Dataset	No.	No. COVID-19 class	% COVID-19 class
All epochs	49250	26116	53.0%
N1 epochs	4668	2996	64.2%
N2 epochs	19708	10516	53.4%
N3 epochs	9844	4612	46.9%
REM epochs	10368	5628	54.3%
Wake epochs	3540	2240	63.3%

Abbreviations: REM, rapid eye movement.

number of features extracted was 20 where most of the features were from the time domain except two that were from the frequency domain.

2.4.1.1 Time Domain Features

Time domain features are descriptive of the morphology of the signal in real time and are usually statistical measures. The time domain features extracted from the signals were physical high, physical low, mean, standard deviation and variance, median, Fisher-Pearson coefficient of skewness and Fisher kurtosis of each epoch. All time domain features were calculated using the Python packages *numPy* [72] and *sciPy* [73].

Signal envelopes are commonly used for amplitude analysis of signals [74]. The upper and lower signal envelopes were calculated for each epoch using cubic spline interpolation and the same time domain features were calculated for both envelopes. Cubic spline interpolation is a method of fitting a set of continuous cubic polynomials to a set of points. In the case of upper and lower signal envelopes the points to be fitted are the highest and lowest points, respectively, of the signal in a particular window [75].

2.4.1.2 Frequency Domain Features

The frequency domain feature extracted was the Power Spectral Density(PSD) estimation using Welch's method. The method first segments the signal, uses finite Fourier transform to form periodograms of each segment followed by averaging of all the periodograms. The segmentation step is formulated in equation 2.1.

$$x_m(n) \triangleq w(n)x(n+mR) \quad n = 0, 1, \ldots, M-1, \; m = 0, 1, \ldots, K-1 \tag{2.1}$$

Where R is defined as the window hop size, K is the number of segments. Equation 2.2 shows the periodogram for the m-th segment.

$$P_{x_m,M}(\omega_k) = \frac{1}{M}\left|\sum_{n=0}^{N-1} x_m(n)e^{-j2\pi nk/N}\right|^2 \tag{2.2}$$

The formulation for the Welch estimate is shown in Equation 2.3.

$$\hat{S}_x^W(\omega_k) \triangleq \frac{1}{K}\sum_{m=0}^{K-1} P_{x_m,M}(\omega_k) \tag{2.3}$$

The Welch PSD estimate is thus the average of the periodograms over time [76], [77]. The PSD was calculated for each epoch using the Python package *SciPy*. Further, the Absolute Power and Spectral Entropy of each epoch was derived from the PSD. The Absolute Power was calculated using composite Simpson's rule which is an approximation of the area under the PSD graph and is shown in Equation 2.4. The Spectral Entropy is a measure of how uniform or complex the energy distribution of the signal is in the frequency domain [78]. The Spectral Entropy was calculated as the Shannon entropy of the PSD and is defined in Equation 2.5 where p_f is the power at frequency f [47].

$$\int_a^b f(x)dx \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right] \tag{2.4}$$

$$H(f) = \sum_f p_f log\left(\frac{1}{p_f}\right) \tag{2.5}$$

2.4.2 Models for Classification

All models in this thesis are trained and evaluated using the *sklearn* Python package which is a open source tool for machine learning data analysis [79]. In section 2.4.3 the metrics used for model performance evaluation are discussed. Each model has certain hyperparameters that can be used to optimize for best performance. Hyperparameter tuning was done using the function *GridSearchCV* where a grid search of all parameters was performed to determine the parameters that result in the best performance of the models.

2.4.3 Model Performance Evaluation

The performance measures used in this thesis are accuracy, recall, precision, F1-score and area AUC. For a binary classification problem the classes can be represented as positives and negatives. True positives and true negatives represent when the model's prediction is correct. False positives and false negatives are the exact opposite, namely when a feature vector is incorrectly predicted as either positive or negative and the actual class of the feature vector contradicts the prediction. To maximize the performance of model the aim is to minimize the false positives and negatives and therefore maximize the ratio of correctly predicted feature vectors.

Accuracy is defined as the ratio of total true predictions of the total feature vectors in the test set as shown in equation 2.6 [53].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.6}$$

Precision is the ratio of correctly predicted positives of total positive predictions. This metric describes how many of those who were classified as positive are actually positive. Recall is the ratio of correctly classified positives of the total number of observations in the positive class [53]. Precision and Recall are defined as shown in equations 2.7 and 2.8, respectively.

$$Precision = \frac{TP}{TP + FP} \tag{2.7}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.8}$$

F1-score is usually a more useful metric than accuracy, especially in cases where the distribution of classes is uneven [53]. The F1-score is based on Precision and Recall as can be seen in Equation 2.9.

$$F1score = \frac{2 \cdot (Precision + Recall)}{Precision + Recall} \tag{2.9}$$

AUC is a performance measure based on the Receiver Operating Characteristic (ROC) curve which is the Recall(True Positive Rate) plotted against the False Positive Rate (FPR). The formula for FPR is shown in Equation 2.10. The AUC is an approximation of the area under the ROC curve [54] where random guessing has the AUC of 0.5 or 50%.

$$FPR = \frac{FP}{FP + TN} \tag{2.10}$$

2.4.3.1 Feature Importance

Feature importance of a model relates to identifying the features in the data that have most impact on the prediction performance [80]. The method used to find the feature importances is called permutation. The Python package *sklearn* offers a function called *permutation_importance* for computing permutation average feature importances. Permutation is a process where one feature at a time is randomly shuffled and the model subsequently evaluated to look for negative effects on performance. The features that have the biggest negative effect on the performance are the most important features [52].

2.4.4 Hyperparameter Tuning

To identify the hyperparameters that resulted in the best performance of each model a grid search was performed using the function *GridSearchCV*. Before any model fitting the datasets were randomly split into training and test sets and the test set size was chosen to be 20%. Which means that 20% of each dataset is not used in the grid search or to fit the models so that the model's performance can be reliably evaluated by comparing predictions with the actual labels of the test set. The cross validation used in the grid search to monitor performance was 5-fold [81].

The method of cross validation grid search is computationally expensive and therefore it was carried out on the Sleep Revolution computer cluster which reduced the computation time considerably and allowed for a larger parameter grid. However, Random Forest performs best with a large number of trees and MLP performs best with many iterations and hidden layers and for that a grid search is time consuming even though it was performed on a computer cluster. Therefore, instead of using the whole datasets when grid searching for best parameters for Random Forest and MLP a random sample of 1000 was used.

Since this particular classification problem had never been applied to machine learning there were no good guidelines for choosing hyperparameters for the models. The hyperparameters that were considered to be most important to each model and likely to have the most effect on the performance were included in the grid search. The values for each parameter in the grid were chosen to be of an adequately wide range and based on other similar projects where machine learning is applied to classify EEG signal epochs [48], [82], [83].

Table 2.3: Tree models hyperparameter grid.

Paramater	Model:**Type/[Values]**	**Description**
n_estimators	RF: **[100-300], stepsize: 100**	The number of tree estimators in the model - only applicable in the ensemble tree model Random Forest.
criterion	DT, RF: **Gini Impurity, Entropy**	The impurity measure used as the splitting criterion.
max_depth	DT: **[3-15], stepsize:1** RF: **[50-200], stepsize: 50**	Maximum depth of a tree, limits the number of splits allowed.
min_samples_split	DT, RF: **[20-50], stepsize: 10**	Minimum sample size for a split, if the sample size in the node is lower than this parameter the node becomes a leaf.
max_features	DT, RF: $\sqrt{\textbf{n\_features}}$, $\textbf{log}_\textbf{2}$	Maximum number of features to consider when deciding a split.
min_samples_leaf	DT, RF: **[1, 4, 6, 10]**	Minimum number of samples allowed in a leaf node - stops a split if there are too few samples left in the node.
bootstrap	RF: **True, False**	Decides whether bootstrap samples of the data is used to fit each tree in a forest or the whole dataset is used to fit each tree estimator.

Abbreviations: RF, Random forest; DT, Decision tree.

2.4.4.1 Tree-based Hyperparameters

As previously discussed, Decision Tree and Random Forest models are both tree-based and the hyperparameters used to tune them are the same. Since the Random Forest model is made up of multiple Decision Trees it has a few additional hyperparameters to tune and two were added in the tuning process. The additional parameters for the Random Forest models are the number of tree estimators and whether each tree gets a random sample as input (bootstrapping). Table 2.3 lists and explains the hyperparameters used to tune the tree-based models and the parameter grid values.

2.4.4.2 Neural Network based Hyperparameters

Neural Networks are complex models and have several tunable hyperparameters. The main parameters for the MLP relate to the mathematical model and structure of the hidden layer. The number of neurons in the hidden layers, activation function of the neurons and how the optimization of the weights is brought about. The activation functions most commonly used are identity-, logistic sigmoid-, hyperbolic tangent and rectified linear unit (ReLU) function. The functions most useful in binary classification problems are the logistic sigmoid an hyperbolic tangent functions and ReLU is the most diverse of the activation functions and therefore all three are used [84]. The solver parameter decides the numerical method for optimization of the weights and the grid includes two stochastic gradient descent (SGD) methods. A summary of the hyperparameters used to tune the MLP classifier is shown in Table 2.4.

Table 2.4: MLP model hyperparameter grid.

Paramater	Type/[Values]	Description
hidden_layer_sizes	**[(50, 50), (50, 100), (100, 100), (50, 100, 100), (100, 100, 100)]**	The number of hidden layers and neurons in each hidden layer.
activation	**ReLU, Sigmoid, Hyperbolic tan**	The activation function in the hidden layer.
solver	**Adam, SGD**	The solver used for weights optimization.
max_iter	**[1500, 2500]**	Maximum number of iterations - the solver iterates until convergence determined by the tolerance which is $1 \cdot 10^{-4}$.
learning_rate	**Constant, Adaptive**	The learning rate plan for weight updates - either adaptive or constant.
learning_rate_init	**[0.01, 0.1]**	The starting learning rates - step sizes in the SGD weights optimizers.

Abbreviations: ReLU, Rectified linear unit function; SGD, Stochastic gradient descent.

Chapter 3

Results

The cohort's demographic descriptors are summarized in Table 3.1. Of the 24 participants in the final cohort, 11 were male and 13 were female. The mean age was 51 years and mean BMI was 28.1 kg/m^2. Hypertension had been diagnosed in 6 of 24 participants. Four of the 11 participants that got COVID-19 were male. In total, three of the participants that got COVID-19 fulfilled the criteria [85] of having long-term COVID-19 symptoms and of those one was male.

3.1 Statistical Analyses

The paired t-test showed significant difference between studies in only one of the primary variables for the COVID-19 group: Mean oxygen saturation. It showed an increase in mean saturation between the two studies (p=0.04). All other variables, both primary and secondary had p-values above the significance threshold. Table 3.2 shows the results of the paired t-test for primary and secondary variables.

Table 3.3 show the power of the paired t-test for the primary and secondary variables of both groups. Further, the results of the analysis to test for minimum sample size to guarantee 80% power is shown in Table 3.4 for all variables. The only variable that has sufficient power is sleep latency in the control group. However, for the COVID-19 group sleep efficiency would only need two more measurement pairs (N=13) to get 80% power.

3.1.1 Aligned Rank Transform ANOVAs

The ART ANOVAs were performed with regards the both factors, COVID-19 status and measurement number. Subsequently, three ANOVAs were performed on all primary and

Table 3.1: Demographic descriptors of the cohort.

Cohort Descriptors	
% Male	46.0
Age [years]	51.0 ± 14.0
BMI [kg/m²]	28.1 ± 5.8
% Hypertension	17.0

Abbreviations: BMI, Body mass index.

Table 3.2: Results from paired t-test on primary variables for COVID-19 and controls. The p-value under the significance threshold is in bold.

	COVID-19 (N=11)			**Controls (N=13)**		
	$\mu_1 \pm \sigma_1$	$\mu_2 \pm \sigma_2$	**p-value**	$\mu_1 \pm \sigma_1$	$\mu_2 \pm \sigma_2$	**p-value**
Total Sleep Time [min]	363.0 ± 120.6	346.5 ± 82.0	0.69	383.5 ± 63.6	394.7 ± 56.6	0.60
Stage N3 [min]	69.7 ± 41.9	58.4 ± 40.5	0.30	81.9 ± 32.9	75.1 ± 23.5	0.58
Stage REM [min]	76.4 ± 38.1	80.9 ± 33.3	0.78	80.0 ± 27.9	85.3 ± 26.6	0.64
Sleep Efficiency [%]	91.5 ± 6.2	87.6 ± 6.3	0.12	90.6 ± 5.7	90.7 ± 5.3	0.94
Mean Oxygen Saturation [%]	92.7 ± 1.5	93.1 ± 1.4	**0.04**	93.5 ± 1.6	93.3 ± 2.0	0.55
Mean Pulse [bpm]	66.1 ± 11.4	64.3 ± 9.1	0.53	65.8 ± 12.1	66.4 ± 13.9	0.86
Sleep Latency [min]	5.0 ± 7.7	8.7 ± 7.3	0.38	7.8 ± 7.4	3.5 ± 2.7	0.08
Arousal Index [arousals/hour]	2.4 ± 3.5	2.9 ± 4.8	0.75	3.5 ± 4.2	2.3 ± 3.9	0.41
Oxygen Saturation below 90% [min]	5.7 ± 9.2	5.4 ± 9.1	0.84	3.6 ± 7.5	7.6 ± 15.1	0.21
ODI	14.0 ± 11.8	11.5 ± 8.3	0.42	11.1 ± 13.5	9.2 ± 8.7	0.49
AHI	14.2 ± 11.6	11.5 ± 8.5	0.38	12.5 ± 14.2	10.5 ± 10.3	0.49

Symbols: μ_1, mean from first study; σ_1, standard deviation from first study; μ_2, mean from second; studyσ_2, standard deviation from second study.

Abbreviations: REM, rapid eye movement; ODI, Oxygen desaturation index; AHI, Apnea-hypopnea index; bpm, beats per minute.

Table 3.3: Power analysis of paired t-test for primary variables for COVID-19 group and Controls.

Variable	**COVID-19(N=11) [Power]**	**Controls(N=13) [Power]**
Total Sleep Time	0.10	0.13
Stage N3	0.20	0.19
Stage REM	0.08	0.14
Sleep Efficiency	0.73	0.05
Mean Oxygen Saturation	0.17	0.08
Mean Pulse	0.11	0.06
Sleep Latency	0.51	0.97
Arousal Index	0.08	0.24
Oxygen Saturation below 90%	0.05	0.36
ODI	0.17	0.12
AHI	0.19	0.12

Abbreviations: REM, rapid eye movement; ODI, Oxygen desaturation index; AHI, Apnea-hypopnea index.

Table 3.4: Minimum number of samples needed to ensure 80% power of paired t-test on primary variables for COVID-19 group and Controls.

Variable	**COVID-19(N=11) [N samples]**	**Controls(N=13) [N samples]**
Total Sleep Time	164	125
Stage N3	59	77
Stage REM	269	114
Sleep Efficiency	13	6759
Mean Oxygen Saturation	74	349
Mean Pulse	141	1906
Sleep Latency	20	8
Arousal Index	328	56
Oxygen Saturation below 90%	4276	36
ODI	75	148
AHI	63	157

Abbreviations: REM, rapid eye movement; ODI, Oxygen desaturation index; AHI, Apnea-hypopnea index.

Table 3.5: Results from three ANOVAs performed on all variables with regards to factors COVID-19, measurement and both COVID-19 and measurement factors together.

Variable	Factor	F-value	p-value
Total Sleep Time [min]	COVID-19	0.72	0.40
	Measurement	0.34	0.56
	COVID-19:Measurement	<0.01	0.96
Stage N3 [min]	COVID-19	1.34	0.25
	Measurement	0.62	0.43
	COVID-19:Measurement	0.003	0.95
Stage REM [min]	COVID-19	0.03	0.87
	Measurement	0.78	0.38
	COVID-19:Measurement	0.10	0.75
Sleep Efficiency [%]	COVID-19	0.12	0.73
	Measurement	<0.01	0.99
	COVID-19:Measurement	0.01	0.91
Mean Oxygen Saturation [%]	COVID-19	1.44	0.24
	Measurement	0.02	0.90
	COVID-19:Measurement	0.16	0.69
Mean Pulse [bpm]	COVID-19	0.01	0.90
	Measurement	0.09	0.77
	COVID-19:Measurement	0.04	0.84
Sleep Latency [min]	COVID-19	<0.01	0.95
	Measurement	0.78	0.38
	COVID-19:Measurement	0.41	0.53
Arousal Index [arousals/hour]	COVID-19	0.49	0.49
	Measurement	0.44	0.51
	COVID-19:Measurement	0.19	0.67
Oxygen Saturation below 90% [min]	COVID-19	0.93	0.34
	Measurement	0.06	0.80
	COVID-19:Measurement	0.09	0.76
ODI	COVID-19	2.08	0.16
	Measurement	0.02	0.89
	COVID-19:Measurement	0.03	0.86
AHI	COVID-19	0.76	0.39
	Measurement	<0.01	0.98
	COVID-19:Measurement	<0.01	0.97

Abbreviations: REM, rapid eye movement; ODI, Oxygen desaturation index; AHI, Apnea-hypopnea index; bpm, beats per minute.

secondary variables: COVID-19 main effect, measurement main effect and interaction effect of both COVID-19 and measurement factors. No significant difference was found in terms of main or interaction effects of the factors. A possible trend towards decrease can be seen in ODI with regards to the COVID-19 factor. Table 3.5 shows the result of the three ART ANOVAs for the primary and secondary variables.

3.2 Machine Learning Methods

In this section the results of the machine learning methods are presented. The machine learning models were tested on six datasets, five different datasets made up of epochs from the different sleep stages (N1, N2, N3, REM, Wake) and also the whole dataset with all epochs from all sleep stages. In the following six sections the performance evaluation of the three types of models, Desicion Tree, Random Forest and MLP on all six datasets are presented along with confusion matrices and seven largest permutation feature importances of the best performing model. The hyperparameters that resulted in the best performance of the models for the six datasets can be found in Appendix A. All Python functions used in signal processing, feature extraction and to train and test the machine learning methods can be found in Appendix B.

3.2.1 Whole Dataset

The performance metrics for the three models trained with the whole dataset, consisting of 49250 feature vectors are shown in Figure 3.1. The benchmark for accuracy was 53%, that is if randomly sampled the likelihood of the sample being from the COVID-19 class is 53%. The Random Forest model, when fitted to the whole dataset, had the overall highest performance metrics of the three models tested, with the Decision Tree a close second and MLP performs worse than both tree based models. The Random Forest model had 64.1% accuracy, 64.9% recall, 66.3% precision, 65.6% F1-score and 64.0% AUC. The parameters that resulted in the best performance when the models were fitted to the whole dataset are shown in Table A.1 in Appendix A. The confusion matrices for the three models can be seen in Figure 3.2. The confusion matrices show a prediction bias towards the COVID-19 class as the majority of the predictions of all models were for the COVID-19 class. For the best performing model, Random Forest, the most important feature was lower envelope standard deviation and the second most important feature was upper envelope standard deviation. Figure 3.3 shows the mean importance of the seven most important features of the Random Forest model when fitted to the whole dataset.

3.2.2 N1 Dataset

The N1 dataset is the second smallest dataset with 4668 feature vectors and 64.2 % belong to the COVID-19 class which makes it the most imbalanced dataset. When tested all models had similar performance measures with Random Forest performing slightly better in all metrics. The performance metrics of all three models can be seen in Figure 3.4. The AUC of the Random Forest, was 64.2% but around 60% for both Decision Tree and MLP. The confusion matrices in Figure 3.5 show that the majority of predictions for all the models is the COVID-19 class which consequently results in high recall for all models, especially for Random Forest and MLP. The mean feature importances of the seven most important features for the best performing model, Random Forest, are shown in Figure 3.6. The most important feature was epoch maximum with almost double the importance of the second most important feature, epoch minimum. Excluding the most important feature, epoch maximum, the six other features have similar mean importances. The best hyperparameters that resulted in the best performance for the three models are shown in Table A.2 in Appendix A.

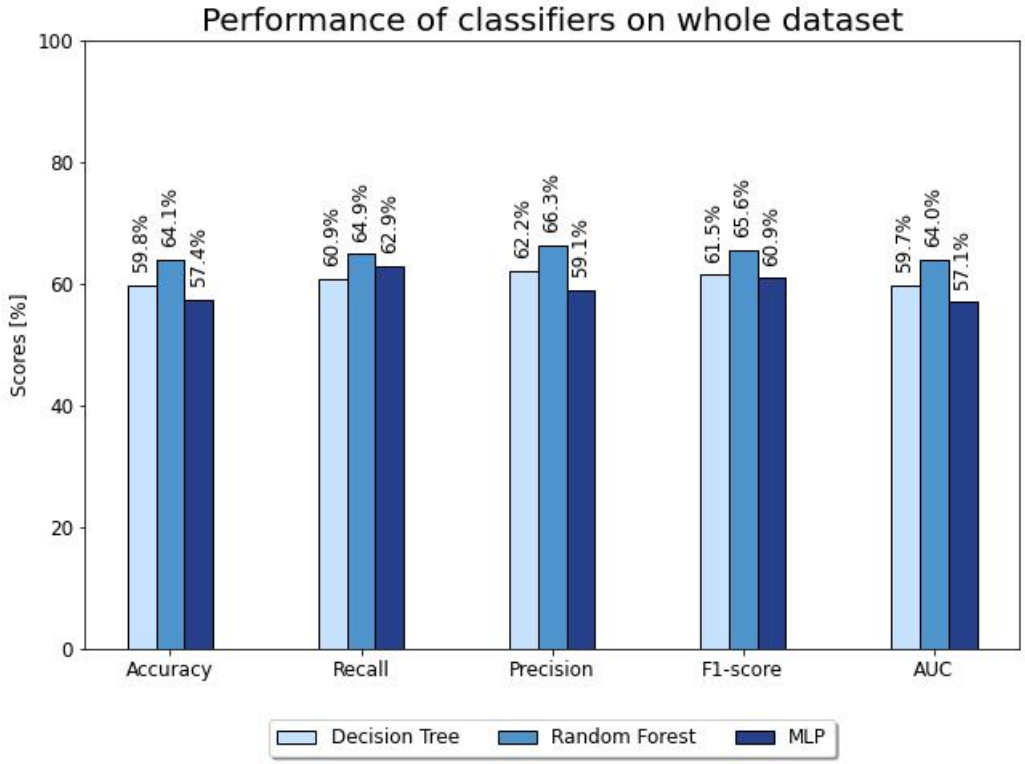


Figure 3.1: Performance statistics for Decision Tree, Random Forest and Multilayer perceptron classifiers trained on the whole dataset.

3.2.3 N2 Dataset

The N2 dataset is the largest of the sleep stage datasets and includes almost half of the whole dataset, that is 19708 feature vectors. The class distribution is evenly balanced with 53.4% in the COVID-19 class. After being fitted to the N2 dataset all the models performed similarly in all metrics as seen in Figure 3.7. Nevertheless Random Forest had the highest performance in all metrics except recall which was 65% and was best for the Decision Tree. The best performing model, Random Forest had 65.2% accuracy, 64.8% recall, 67.8% precision, 66.3% F1-score and 65.2% AUC. The parameter grids used to tune the three models are seen in Table A.3 in Appendix A. The confusion matrices show that all models have a similar number of TP, but the Random Forest has the highest number of TN which results in the slightly better performance metrics. Figure 3.8 shows the confusion matrices for the three models. The seven most important features of the Random Forest, the best performing model, fitted to the N2 dataset can be seen in Figure 3.9. The most important feature was upper envelope skew and epoch minimum, but all seven most important features have similar mean importance, that is many features have similar effect on the predictions of the model.

3.2.4 N3 Dataset

The N3 dataset consist of 9844 feature vectors and is adequately balanced with 46.9% belonging to the COVID-19 class. Figure 3.10 shows the performance metrics for the three models when fitted to the N3 dataset. The Random Forest classifier had the highest performance in all metrics when fitted to the N3 dataset. It has 71.8% accuracy, 59.1% recall, 75.9% precision, 66.4% F1-score and 71.1% AUC. Decision Tree and MLP had

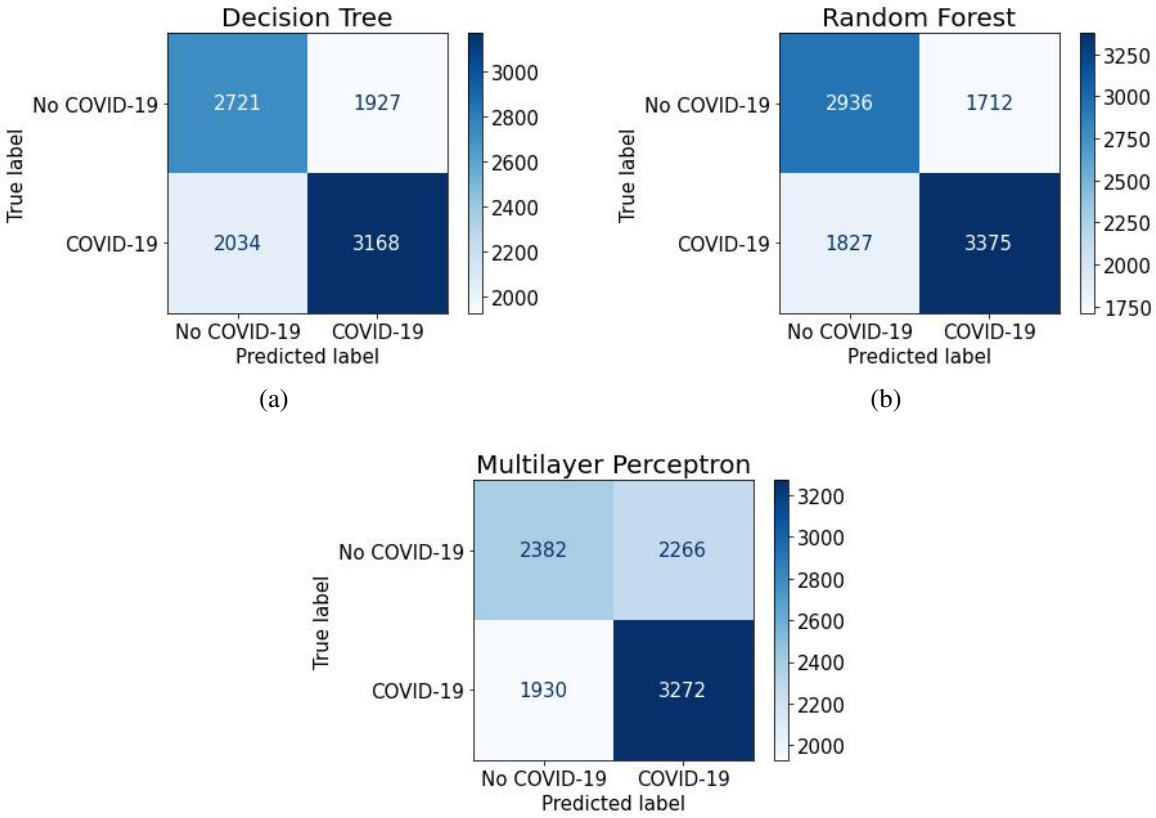


Figure 3.2: Confusion matrices for the three models tested on the whole dataset.

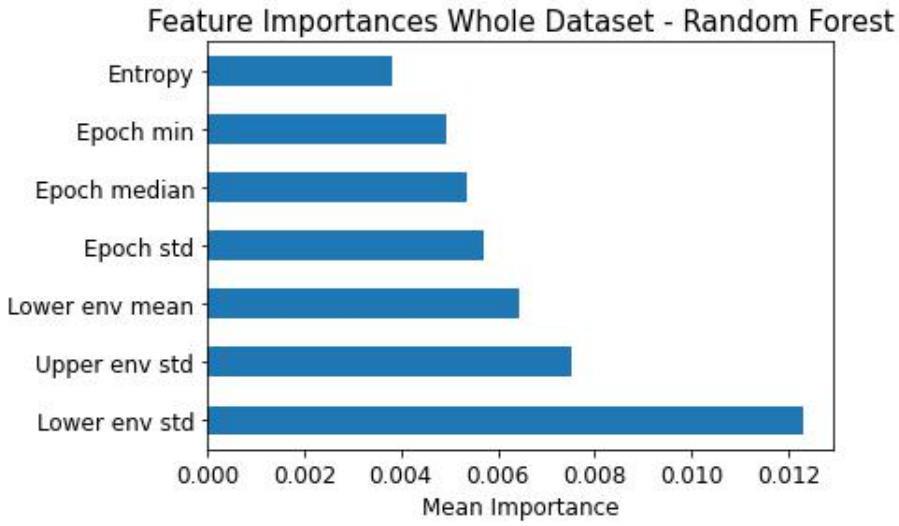


Figure 3.3: Seven highest feature importances of 20 features for the best performing model, Random Forest, fitted to the whole dataset.

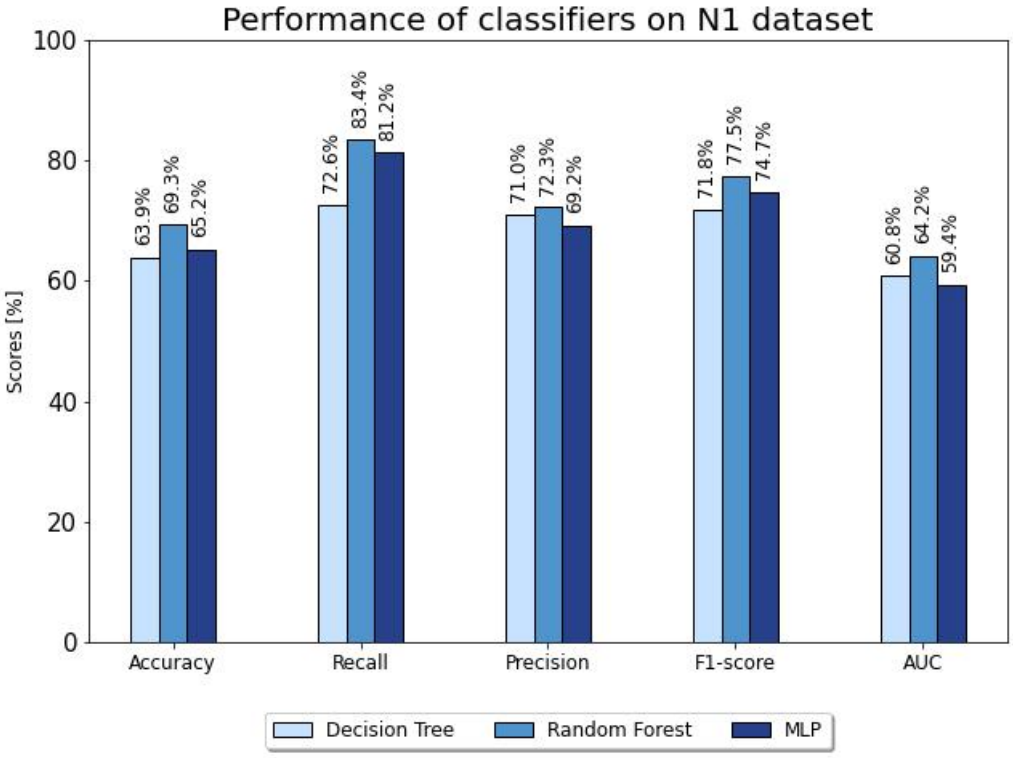


Figure 3.4: Performance statistics for Decision Tree, Random Forest and Multilayer perceptron classifiers trained on the dataset consisting of N1 epochs.

similar and slightly lower performance metrics compared to the Random Forest. The MLP has the lowest performance metrics, particularly the 50.2% recall. The confusion matrices in Figure 3.11 indicate a prediction bias toward the No-COVID-19 class for all of the models but the more severe bias seen in the Decision Tree and MLP models. The largest feature importance for the best performing model, Random Forest, was epoch minimum. The second most important feature, with similar importance, was epoch standard deviation. The 7 most important features for the Random Forest models can be seen in Figure 3.12. The parameter grids used to tune the three models on the N3 dataset are shown in Table A.4 in Appendix A.

3.2.5 REM Dataset

The REM dataset consists of 10368 feature vectors and 54.3% are in the COVID-19 class. When trained on the REM dataset all classifiers had similar performance but the Random Forest classifier had better overall metrics. It had 66.0% accuracy, 69.8% recall, 69.9% precision, 69.8% F1 score and 65.4% AUC. The performances of all three models are displayed in Figure 3.13. The confusion matrices for the three models are shown in Figure 3.14. The confusion matrices show that the majority prediction for all models was the COVID-19 class. The Random Forest hast the highest number of TP and TN which results in the best overall metrics. The Decision Tree had the highest number of FP and MLP had the highest number of FN. The feature importances reveal that the most important feature of the Random Forest model was entropy with more than double the mean importance of the second most important feature, epoch maximum. The seven most important features of the Random Forest model can be seen in Figure 3.15. The hyperparameters that resulted in the best performance of the models are in Table A.5 in Appendix A.

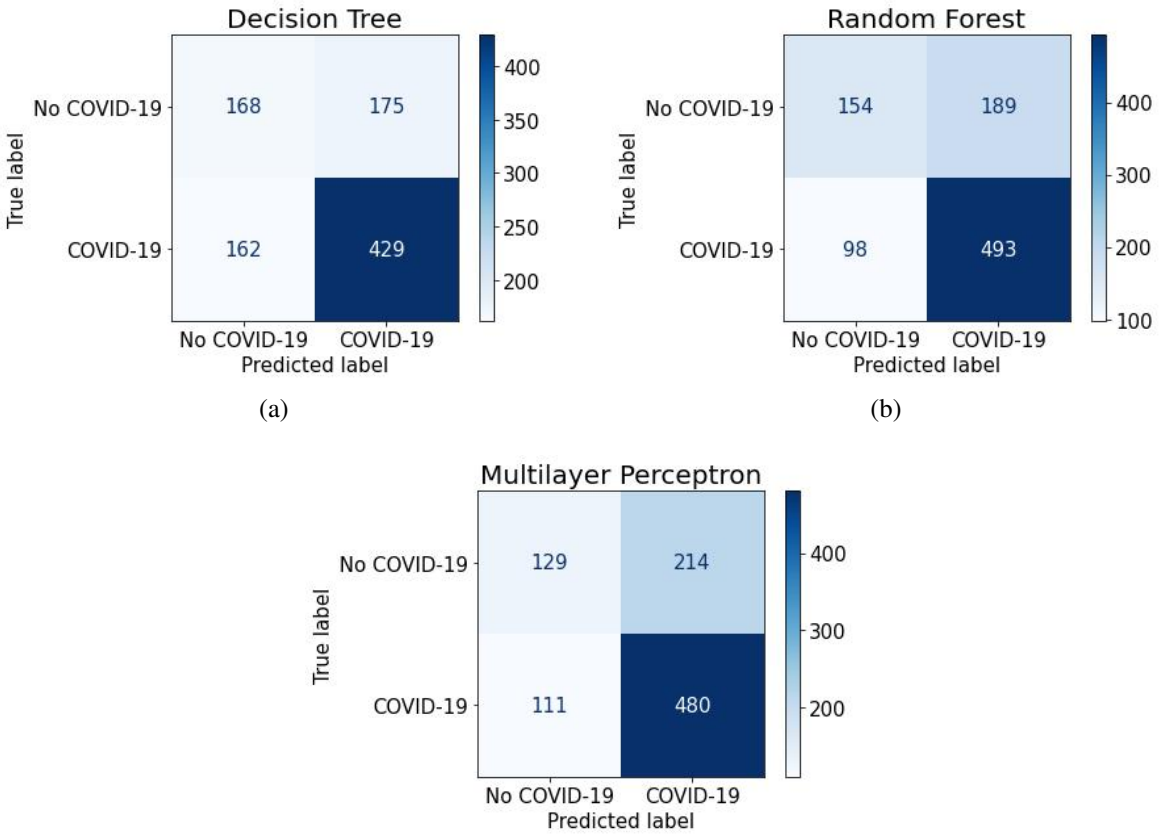


Figure 3.5: Confusion matrices for the three models tested on the N1 dataset.

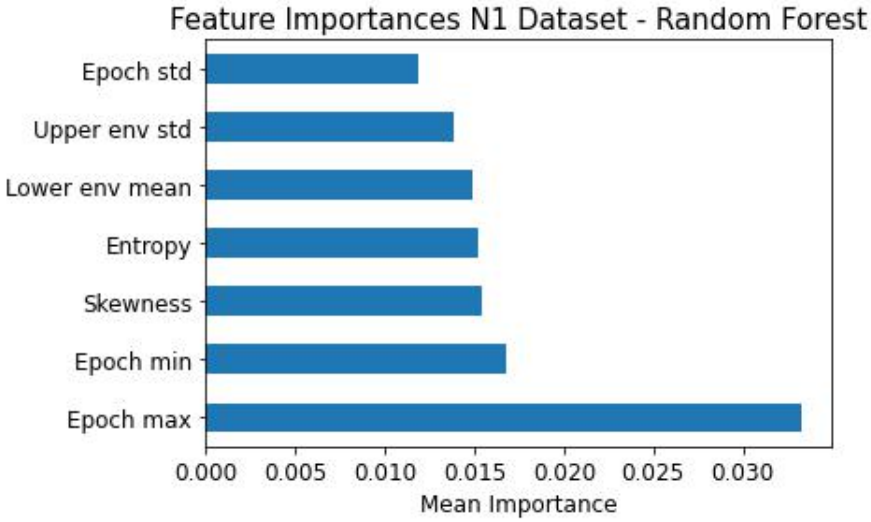


Figure 3.6: Seven highest feature importances of 20 features for the best performing model, Random Forest, fitted to the N1 dataset.

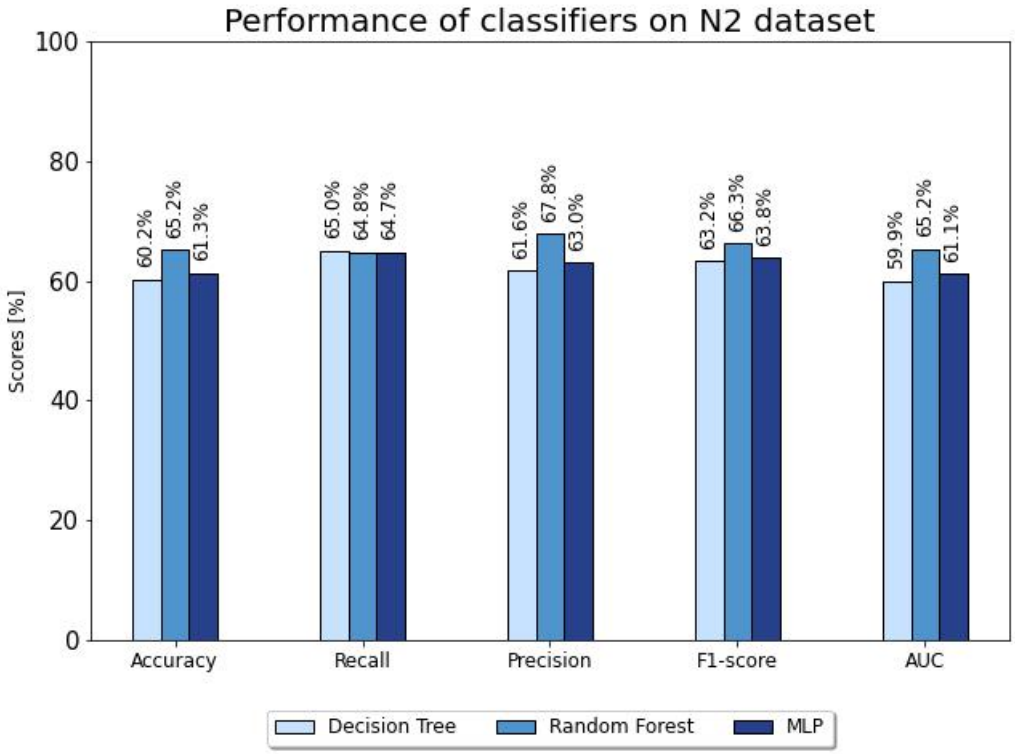


Figure 3.7: Performance statistics for Decision Tree, Random Forest and Multilayer perceptron classifiers trained on the dataset consisting of N2 epochs.

3.2.6 Wake Dataset

The Wake dataset is the smallest dataset with 3540 feature vectors and it has the second most class imbalance of all the datasets with the majority class being COVID-19 with 63.3% of the feature vectors in the set. The performance metrics of the three models when trained and tested on the Wake dataset can be seen in Figure 3.16. The Random Forest has the best performance according to all metrics. The Random Forest has 73.7% accuracy, 84.7% recall, 76.5% precision, 80.4% F1-score and 69.6% AUC. The confusion matrices in Figure 3.17 indicate that all three models show a prediction bias towards the COVID-19 class where MLP has the most severe bias. The confusion matrix of the Random Forest model shows how it had the highest number of both TP and TN and therefore the best performance. The parameters that resulted in the best performance when the models are trained on the Wake dataset can be seen in Table A.6 in Appendix A. The seven most important features for the best performing model are shown in Figure 3.18. The most important features for the Random Forest classifier is entropy. Two of the seven most important features have almost negligible mean importance.

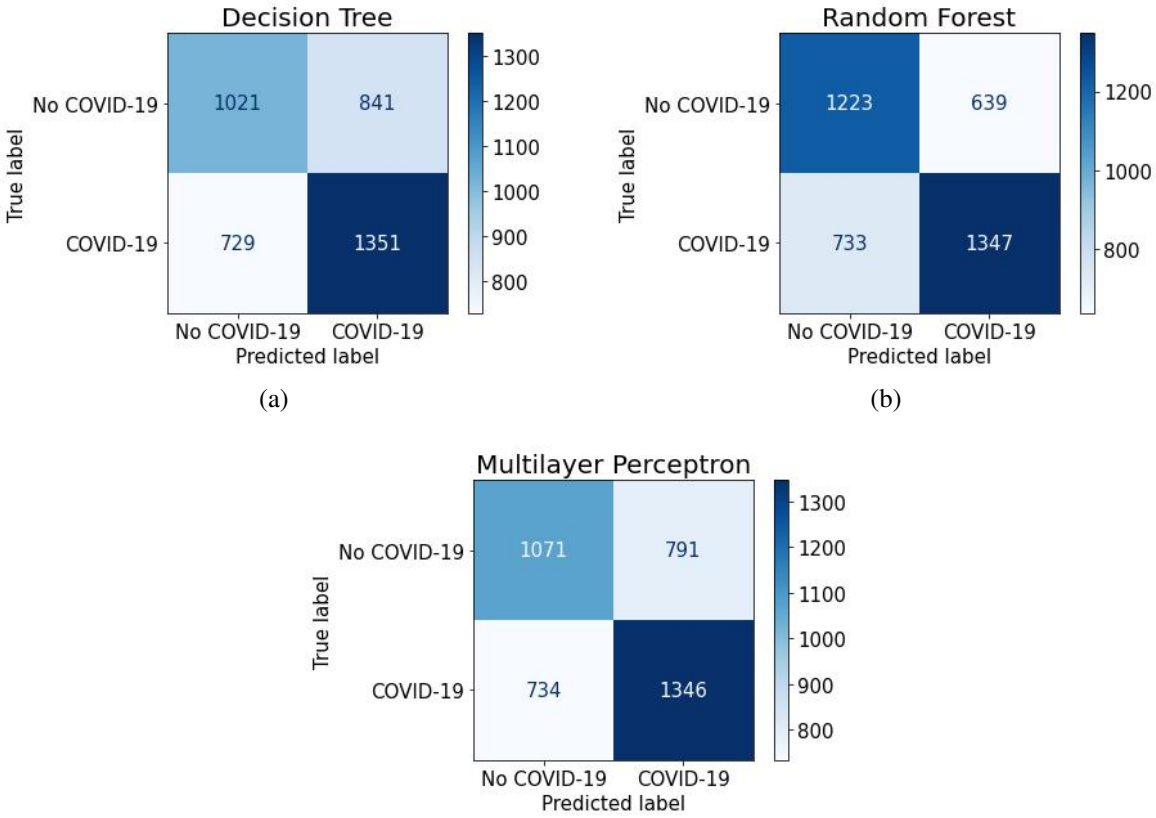


Figure 3.8: Confusion matrices for the three models tested on the N2 dataset.

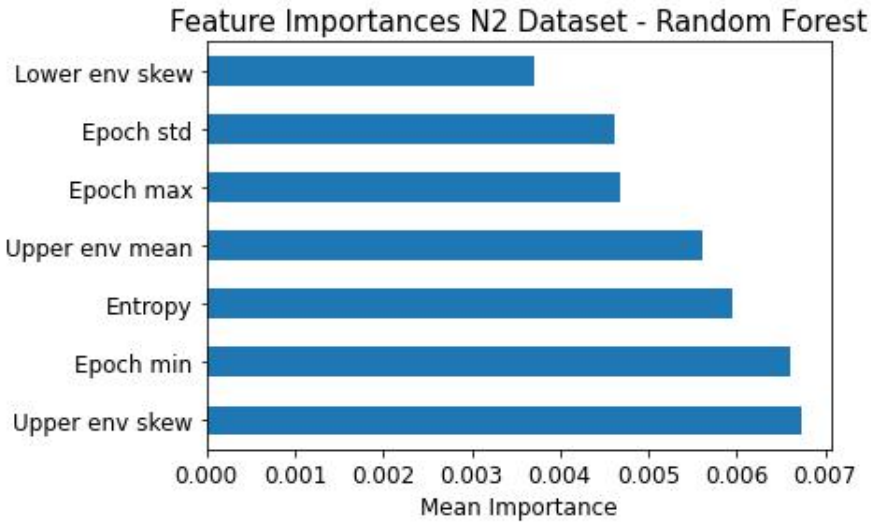


Figure 3.9: Seven highest feature importances of 20 features for the best performing model, Random Forest, fitted to the N2 dataset.

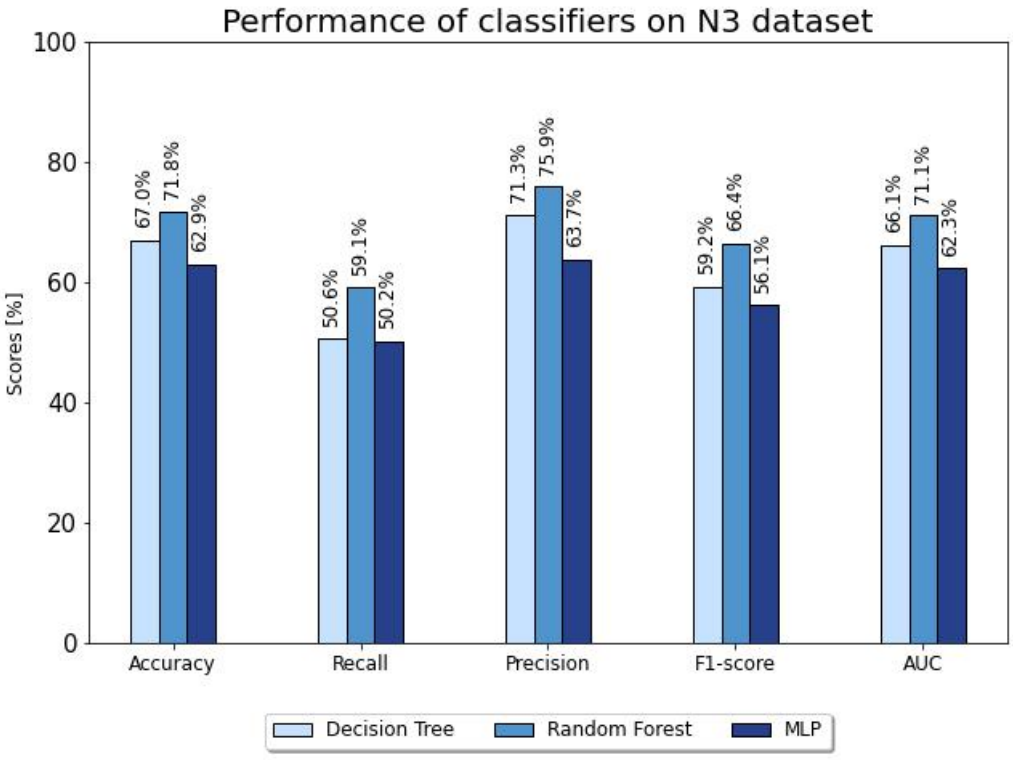


Figure 3.10: Performance statistics for Decision Tree, Random Forest and Multilayer Perceptron classifiers trained on the dataset consisting of N3 epochs.

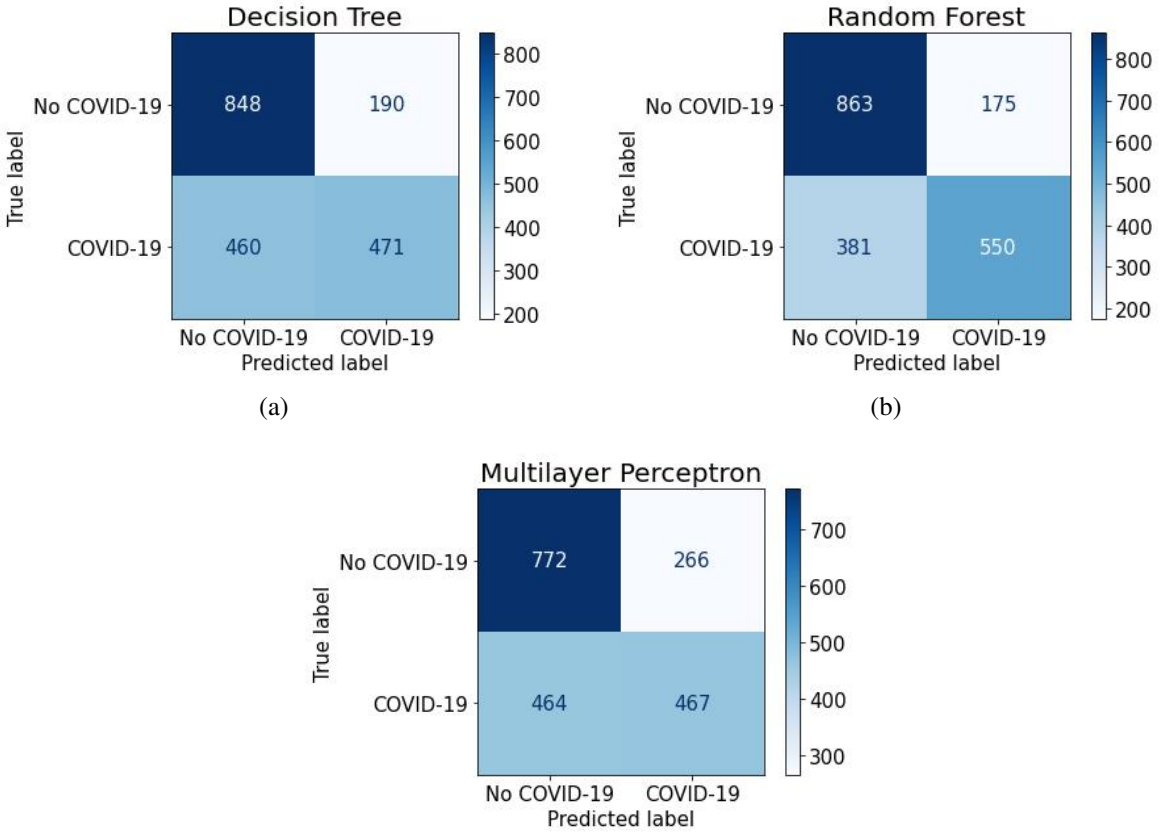


Figure 3.11: Confusion matrices for the three models tested on the N3 dataset.

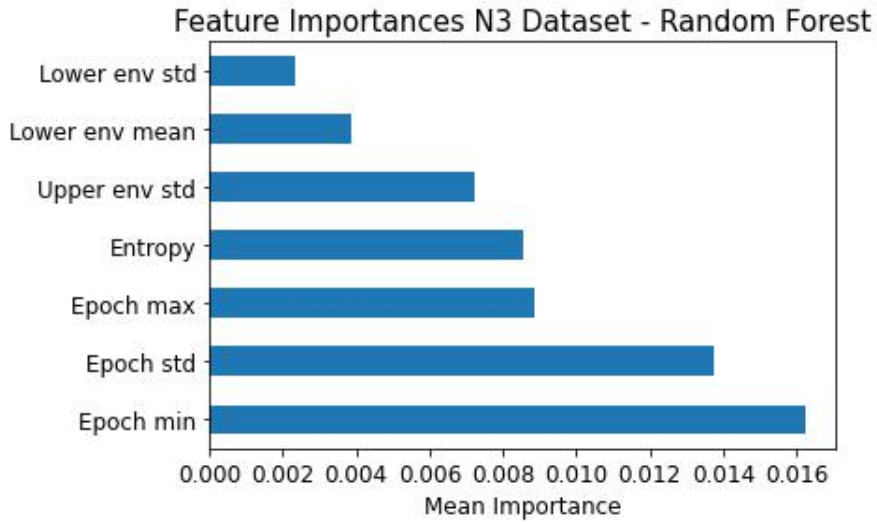


Figure 3.12: Seven highest feature importances of 20 features for the best performing model, Random Forest, fitted to the N3 dataset.

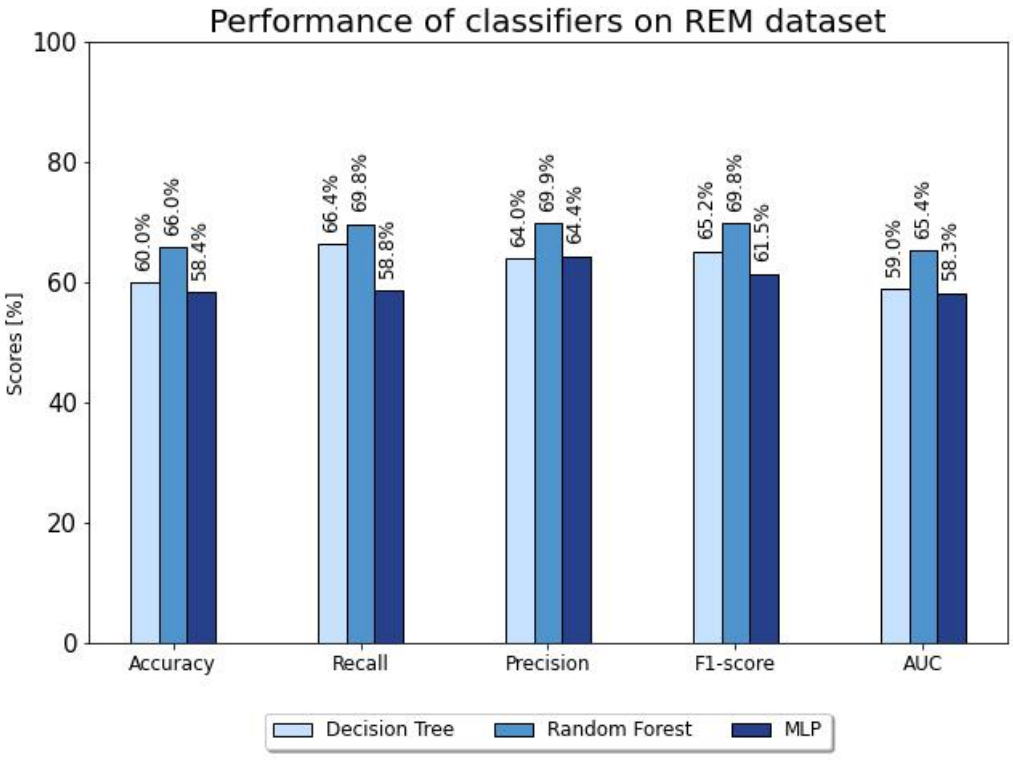


Figure 3.13: Performance statistics for Decision Tree, Random Forest and Multilayer Perceptron classifiers trained on the dataset consisting of REM epochs.

Figure 3.14: Confusion matrices for the three models tested on the REM dataset.

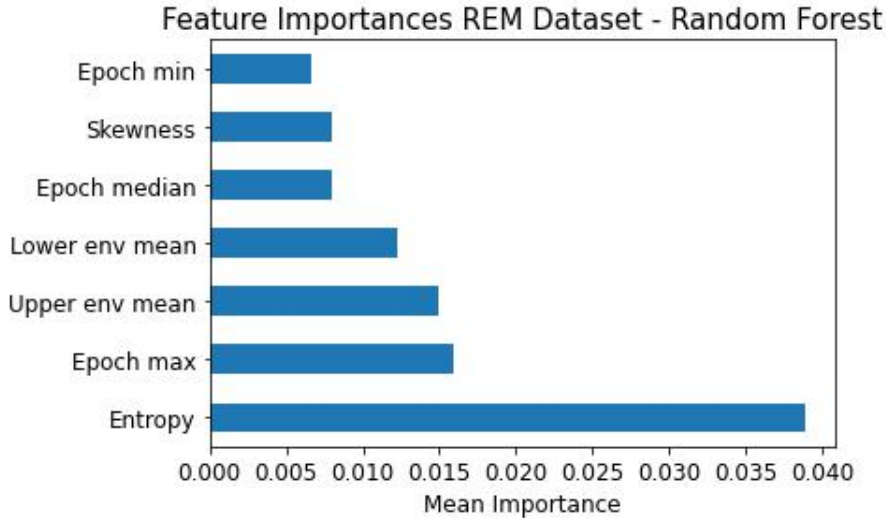


Figure 3.15: Seven highest feature importances of 20 features for the best performing model, Random Forest, fitted to the REM dataset.

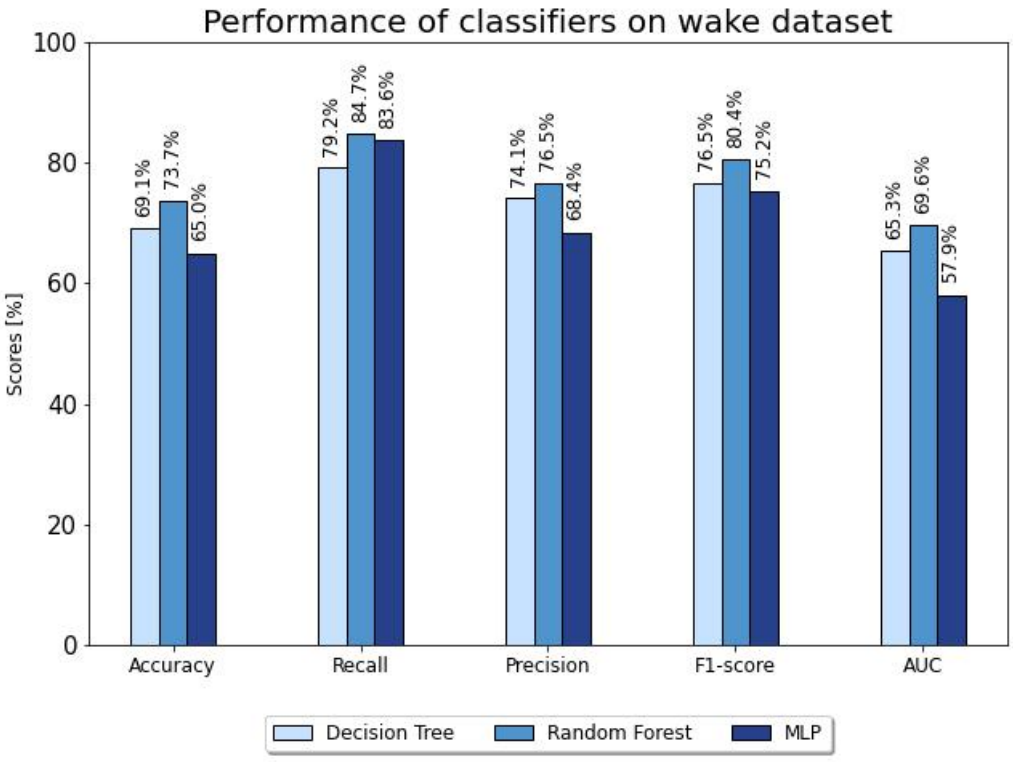
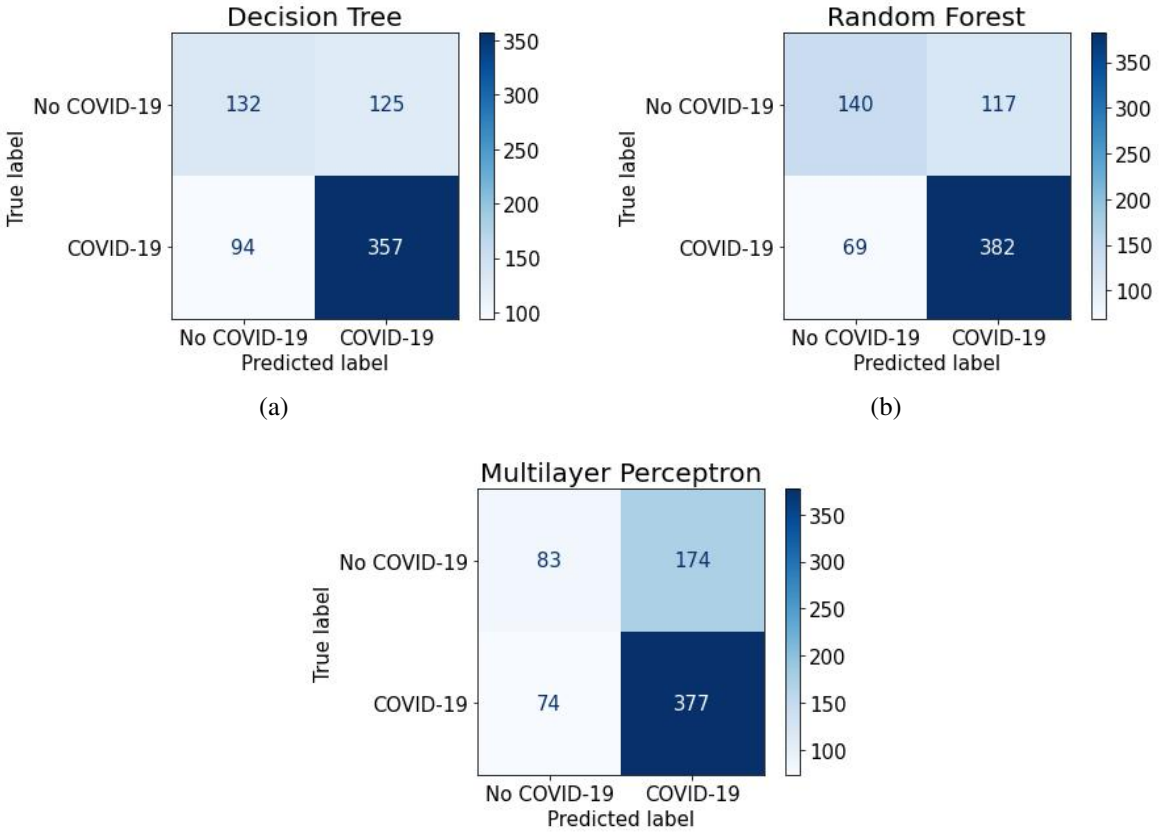


Figure 3.16: Performance statistics for Decision Tree, Random Forest and Multilayer perceptron classifiers trained on the dataset consisting of Wake epochs.



(a) (b)

(c)

Figure 3.17: Confusion matrices for the three models tested on the wake dataset.

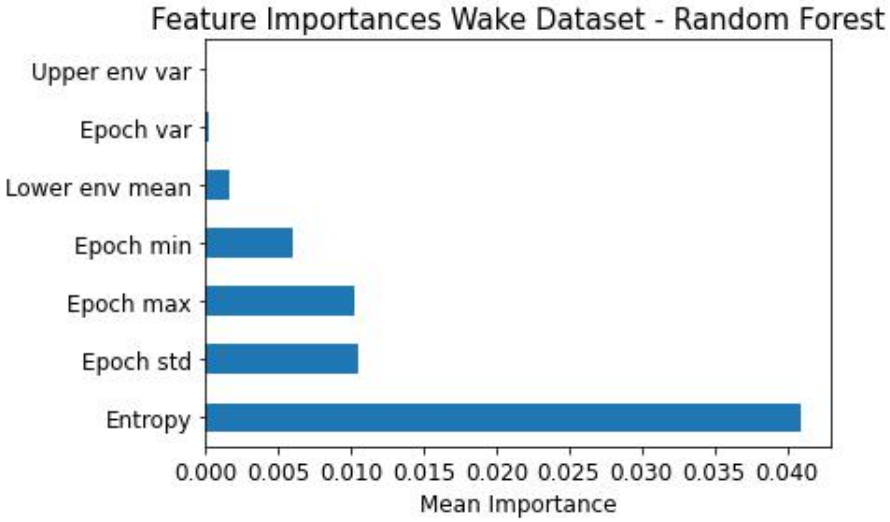


Figure 3.18: Seven highest feature importances of 20 features for the best performing model fitted to the Wake dataset.

Chapter 4

Discussion

No significant effects of COVID-19 were found on clinical objective sleep quality parameters for any of the variables tested, except for mean saturation in the COVID-19 group where the oxygen saturation was slightly elevated between the two studies. This elevation is in direct contrast to what was hypothesized as COVID-19 is primarily a respiratory infection and can damage the lungs of the host [3], [86]. This might indicate that the change is due to other factors such as weight loss or another sporadic effect.

Due to how small the dataset was, it was argued that the paired t-test would not be powerful enough to detect differences and therefore, a power analysis was conducted. The power analysis revealed insufficient power to detect a difference in all primary and secondary variables for both COVID-19 subjects and controls, with the exception of sleep latency for the control group. As mentioned, the main reason for this power deficiency is likely to be the smallness of the dataset which consequently results in small effect size. Additionally, a power analysis was performed to investigate how many more subjects would be needed to ensure 80% power. Sleep efficiency in the COVID-19 group would have needed two more measurement pairs to elevate the power to 80%. However, large clinically meaningful differences could be visible at least as a trend in the data as seen in sleep efficiency of the COVID-19 group and sleep latency for the controls group.

The non-parametric analysis alternative was ART ANOVA which revealed no significant difference in terms of any of the factors, that is no differences were significant in terms of the main effects of the COVID-19 factor or measurement factor, nor the interaction effect between the COVID-19 and measurement factors. However, a possible trend towards decrease in ODI can be seen in terms of the COVID-19 factor. These findings indicate that it cannot be stated with any confidence that COVID-19 affects sleep quality in terms of the primary and secondary variables evaluated, according to the tests conducted as primary objective of this thesis.

The secondary objectives were to use three supervised machine learning methods, Decision Tree, Random Forest and MLP, to investigate whether it was possible to predict with adequate confidence whether epoch features from different sleep stages were from a recording before or after contracting COVID-19. The overall best performing dataset was the N3 dataset for the Random Forest classifier. The Random Forest was the overall best performing classifier with AUC consistently 10% above and up to 21% above random guessing for all datasets. The second best performance of the Random Forest classifier was when fitted to the dataset consisting of wake epochs, where the AUC was around 69%. Change has been described in sleep patterns after viral infections, especially in the REM and N3 stages which could be the reason for the performance results for the models when fitted to the REM and N3 datasets [40]–[42]. The good performance of the wake dataset relative to the other

datasets (N1, N2) might be due to changes in the EEG activity that causes arousals and wake periods during the night. Although up to 21% over random guessing is not considered high performance by any measure, the fact that all the models consistently did better than random guessing, one can postulate that some changes might be present in EEG morphology and therefore it is worth further investigation.

The reasons for the performance inadequacies could lie either in the dataset, the models or both. The datasets consisted of features from 30 second epochs of signals sampled at 200 Hz. It can be speculated that the features extracted were weak in terms of detecting differences and other features might be better suited for detecting differences caused by COVID-19. According to the feature importances the most important features for the best performing models were frequently from the time domain and spectral entropy was the frequency domain feature that consistently was among the seven most important features. Therefore, these features need further analysis in future studies of these datasets and others.

4.1 Future Work

When working with small datasets the problem often is power of analysis, which means being able to detect an effect if it exists. The obvious answer to this problem would be to gather enough data to result in at least 80% power for clinically meaningful differences. However, that is not always possible and therefore finding analyses where power is higher for a small dataset should be an option to consider.

The next steps in the machine learning analysis would be to investigate the possible causes for the changes detected. In this thesis the data consists of epochs from the same subjects in both classes. By conducting the same experiment on a larger dataset with different subjects in both classes and adding a control group, could confirm or disprove that the changes detected are caused by COVID-19.

Since the features extracted from the EEG signal epochs did not result in robust model performance the next steps could include the addition of more signals from each recording. As discussed in section 2.1, the self applied PSG collects multiple signals from each subject and therefore, instead of solely using the EEG signal more signals could be added to the dataset. The signals that could be added are for example the breathing signals, i.e. oxygen saturation, airflow and breathing belts. Moreover, the movements and EMG and other scored events could further add to the possible performance and robustness of the models.

The 20 features extracted might not be ideal for this task and therefore, adding more features from the frequency and time-frequency domains and using a feature selection method could help boost the performances of the models. Methods such as Wavelet Transform have been used to classify EEG epochs with adequate performance [47], [71], [87] and should be considered in next steps.

Hyperparameter tuning was a limiting factor due to computational time and better performances might be gained by widening the range of parameters and performing the grid search on all samples in each dataset rather than a random subset of 1000.

4.2 Conclusion

The main objectives were to investigate whether COVID-19 affects sleep quality by analysing objective sleep measures. The parametric test revealed that the mean saturation was significantly elevated between studies. Due to insufficient power a non-parametric alternative test

was performed an it did not find any significant difference between the two studies in terms of any of the variables tested, including mean saturation. Therefore, it cannot be stated with any confidence that COVID-19 has negative effects on sleep quality in terms of the variables investigated.

The secondary objectives involved using machine learning methods to detect differences in features from EEG signal epochs of different sleep stages from measurements before and after COVID-19. The results indicate that a slight difference is consistently detected and most prominently in the N3, REM and Wake datasets. Since changes in REM and deep sleep have been described after viral infections, these findings should warrant further research into effects of COVID-19 infections on EEG activity during sleep.

Bibliography

[1] S. S, W. G, S. W, L. J, L. ACK, Z. J, L. W, B. Y, and G. GF, "Epidemiology, Genetic Recombination, and Pathogenesis of Coronaviruses", *Trends in microbiology*, vol. 24, no. 6, pp. 490–502, Jun. 2016, ISSN: 1878-4380. DOI: 10.1016/J.TIM.2016.03.003. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/27012512/.

[2] J. Cui, F. Li, and Z.-L. Shi, "Origin and evolution of pathogenic coronaviruses", *Nature Reviews Microbiology 2018 17:3*, vol. 17, no. 3, pp. 181–192, Dec. 2018, ISSN: 1740-1534. DOI: 10.1038/s41579-018-0118-9. [Online]. Available: https://www.nature.com/articles/s41579-018-0118-9.

[3] L. JR, C.-E. CM, D. S. DR, H. M, L. B. SD, R. A, D. D, B. S, E. A. F, D. L, C.-I. Y, H. S, D. IL, C.-H. C, L. P, F. C, B. MR, C. G, K. M, L. P, S. C, R. C, J. F, H. J, E. M, C. R, R. L, L. A, D. F. C, C. F, F. N, A. T, and S. S, "Olfactory and gustatory dysfunctions as a clinical presentation of mild-to-moderate forms of the coronavirus disease (COVID-19): a multicenter European study", *European archives of oto-rhino-laryngology : official journal of the European Federation of Oto-Rhino-Laryngological Societies (EUFOS) : affiliated with the German Society for Oto-Rhino-Laryngology - Head and Neck Surgery*, vol. 277, no. 8, pp. 2251–2261, Aug. 2020, ISSN: 1434-4726. DOI: 10.1007/S00405-020-05965-1. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32253535/.

[4] J. Meinhardt, J. Radke, C. Dittmayer, J. Franz, C. Thomas, R. Mothes, M. Laue, J. Schneider, S. Brünink, S. Greuel, M. Lehmann, O. Hassan, T. Aschman, E. Schumann, R. L. Chua, C. Conrad, R. Eils, W. Stenzel, M. Windgassen, L. Rößler, H.-H. Goebel, H. R. Gelderblom, H. Martin, A. Nitsche, W. J. Schulz-Schaeffer, S. Hakroush, M. S. Winkler, B. Tampe, F. Scheibe, P. Körtvélyessy, D. Reinhold, B. Siegmund, A. A. Kühl, S. Elezkurtaj, D. Horst, L. Oesterhelweg, M. Tsokos, B. Ingold-Heppner, C. Stadelmann, C. Drosten, V. M. Corman, H. Radbruch, and F. L. Heppner, "Olfactory transmucosal SARS-CoV-2 invasion as a port of central nervous system entry in individuals with COVID-19", *Nature Neuroscience 2020 24:2*, vol. 24, no. 2, pp. 168–175, Nov. 2020, ISSN: 1546-1726. DOI: 10.1038/s41593-020-00758-5. [Online]. Available: https://www.nature.com/articles/s41593-020-00758-5.

[5] A. M. Baig, A. Khaleeq, U. Ali, and H. Syeda, "Evidence of the COVID-19 Virus Targeting the CNS: Tissue Distribution, Host–Virus Interaction, and Proposed Neurotropic Mechanisms", *ACS Chemical Neuroscience*, vol. 11, no. 7, pp. 995–998, Apr. 2020. DOI: 10.1021/ACSCHEMNEURO.0C00122. [Online]. Available: https://pubs.acs.org/doi/full/10.1021/acschemneuro.0c00122.

[6] L. YC, B. WZ, and H. T, "The neuroinvasive potential of SARS-CoV2 may play a role in the respiratory failure of COVID-19 patients", *Journal of medical virology*, vol. 92, no. 6, pp. 552–555, Jun. 2020, ISSN: 1096-9071. DOI: 10.1002/JMV.25728. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32104915/.

[7] A. S. Zubair, L. S. McAlpine, T. Gardin, S. Farhadian, D. E. Kuruvilla, and S. Spudich, "Neuropathogenesis and Neurologic Manifestations of the Coronaviruses in the Age of Coronavirus Disease 2019: A Review", *JAMA Neurology*, vol. 77, no. 8, pp. 1018–1027, Aug. 2020, ISSN: 2168-6149. DOI: 10.1001/JAMANEUROL.2020.2065. [Online]. Available: https://jamanetwork.com/journals/jamaneurology/fullarticle/2766766.

[8] G. C. Cardona, L. D. Q. Pájaro, I. D. Q. Marzola, Y. R. Villegas, and L. R. M. Salazar, "Neurotropism of SARS-CoV 2: Mechanisms and manifestations", *Journal of the Neurological Sciences*, vol. 412, p. 116 824, May 2020. DOI: 10.1016/J.JNS.2020.116824. [Online]. Available: /pmc/articles/PMC7141641/%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7141641/.

[9] G. Tsivgoulis, L. Palaiodimou, A. H. Katsanos, V. Caso, M. Köhrmann, C. Molina, C. Cordonnier, U. Fischer, P. Kelly, V. K. Sharma, A. C. Chan, R. Zand, A. Sarraj, P. D. Schellinger, K. I. Voumvourakis, N. Grigoriadis, A. V. Alexandrov, and S. Tsiodras, "Neurological manifestations and implications of COVID-19 pandemic", *Therapeutic Advances in Neurological Disorders*, vol. 13, 2020. DOI: 10.1177/1756286420932036.

[10] A. I and R. FA, "Neurological manifestations and complications of COVID-19: A literature review", *Journal of clinical neuroscience : official journal of the Neurosurgical Society of Australasia*, vol. 77, pp. 8–12, Jul. 2020, ISSN: 1532-2653. DOI: 10.1016/J.JOCN.2020.05.017. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32409215/.

[11] F. A, S. P, H. F, and E. PS, "Neurological Complications of Coronavirus Disease (COVID-19): Encephalopathy", *Cureus*, vol. 12, no. 3, Mar. 2020, ISSN: 2168-8184. DOI: 10.7759/CUREUS.7352. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32328364/.

[12] P. Gklinos, "Neurological manifestations of COVID-19: a review of what we know so far", *Journal of Neurology*, vol. 267, no. 9, p. 1, Sep. 2020. DOI: 10.1007/S00415-020-09939-5. [Online]. Available: /pmc/articles/PMC7250266/%20/pmc/articles/PMC7250266/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7250266/.

[13] N. HR, Z. B, N. A, and B. N, "The neurological manifestations of COVID-19: a review article", *Neurological sciences : official journal of the Italian Neurological Society and of the Italian Society of Clinical Neurophysiology*, vol. 41, no. 7, pp. 1667–1671, Jul. 2020, ISSN: 1590-3478. DOI: 10.1007/S10072-020-04486-3. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32483687/.

[14] A. O. Correia, P. W. G. Feitosa, J. L. d. S. Moreira, S. Á. R. Nogueira, R. B. Fonseca, and M. E. P. Nobre, "Neurological manifestations of COVID-19 and other coronaviruses: A systematic review", *Neurology Psychiatry and Brain Research*, vol. 37, pp. 27–32, Sep. 2020. DOI: 10.1016/J.NPBR.2020.05.008.

[15] S. Kremer, F. Lersy, J. d. Sèze, J.-C. Ferré, A. Maamar, B. Carsin-Nicol, O. Collange, F. Bonneville, G. Adam, G. Martin-Blondel, M. Rafiq, T. Geeraerts, L. Delamarre, S. Grand, A. Krainik, S. Caillard, J. M. Constans, S. Metanbou, A. Heintz, J. Helms, M. Schenck, N. Lefèbvre, C. Boutet, X. Fabre, G. Forestier, I. d. Beaurepaire, G. Bornet, A. Lacalm, H. Oesterlé, F. Bolognini, J. Messié, G. Hmeydia, J. Benzakoun, C. Oppenheim, B. Bapst, I. Megdiche, M.-C. H. Feugeas, A. Khalil, A. Gaudemer,

L. Jager, P. Nesser, Y. T. Mba, C. Hemmert, P. Feuerstein, N. Sebag, S. Carré, M. Alleg, C. Lecocq, E. Schmitt, R. Anxionnat, F. Zhu, P.-O. Comby, F. Ricolfi, P. Thouant, H. Desal, G. Boulouis, J. Berge, A. Kazémi, N. Pyatigorskaya, A. Lecler, S. Saleme, M. Edjlali-Goujon, B. Kerleroux, P.-E. Zorn, M. Matthieu, S. Baloglu, F.-D. Ardellier, T. Willaume, J. C. Brisset, C. Boulay, V. Mutschler, Y. Hansmann, P.-M. Mertes, F. Schneider, S. Fafi-Kremer, M. Ohana, F. Meziani, J.-S. David, N. Meyer, M. Anheim, and F. Cotton, "Brain MRI Findings in Severe COVID-19: A Retrospective Observational Study", *https://doi.org/10.1148/radiol.2020202222*, vol. 297, no. 2, E242–E251, Jun. 2020. DOI: 10.1148/RADIOL.2020202222. [Online]. Available: https://pubs.rsna.org/doi/abs/10.1148/radiol.2020202222.

[16] J. A. Frontera, D. Yang, A. Lewis, P. Patel, C. Medicherla, V. Arena, T. Fang, A. Andino, T. Snyder, M. Madhavan, D. Gratch, B. Fuchs, A. Dessy, M. Canizares, R. Jauregui, B. Thomas, K. Bauman, A. Olivera, D. Bhagat, M. Sonson, G. Park, R. Stainman, B. Sunwoo, D. Talmasov, M. Tamimi, Y. Zhu, J. Rosenthal, L. Dygert, M. Ristic, H. Ishii, E. Valdes, M. Omari, L. Gurin, J. Huang, B. M. Czeisler, D. E. Kahn, T. Zhou, J. Lin, A. S. Lord, K. Melmed, S. Meropol, A. B. Troxel, E. Petkova, T. Wisniewski, L. Balcer, C. Morrison, S. Yaghi, and S. Galetta, "A prospective study of long-term outcomes among hospitalized COVID-19 patients with and without neurological complications", *Journal of the Neurological Sciences*, vol. 426, p. 117 486, 2021. DOI: 10.1016/j.jns.2021.117486. [Online]. Available: https://doi.org/10.1016/j.jns.2021.117486.

[17] B. AM, "Deleterious Outcomes in Long-Hauler COVID-19: The Effects of SARS-CoV-2 on the CNS in Chronic COVID Syndrome", *ACS chemical neuroscience*, vol. 11, no. 24, pp. 4017–4020, Dec. 2020, ISSN: 1948-7193. DOI: 10.1021/ACSCHEMNEURO.0C00725. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33275404/.

[18] N. T, H. M, and G. SN, "Assessment of the Frequency and Variety of Persistent Symptoms Among Patients With COVID-19: A Systematic Review", *JAMA network open*, vol. 4, no. 5, May 2021, ISSN: 2574-3805. DOI: 10.1001/JAMANETWORKOPEN.2021.11417. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/34037731/.

[19] S. Lopez-Leon, T. Wegman-Ostrosky, C. Perelman, R. Sepulveda, P. A. Rebolledo, A. Cuapio, and S. Villapol, "More than 50 long-term effects of COVID-19: a systematic review and meta-analysis", *Scientific Reports 2021 11:1*, vol. 11, no. 1, pp. 1–12, Aug. 2021, ISSN: 2045-2322. DOI: 10.1038/s41598-021-95565-8. [Online]. Available: https://www.nature.com/articles/s41598-021-95565-8.

[20] C. A, B. R, and L. F, "Persistent Symptoms in Patients After Acute COVID-19", *JAMA*, vol. 324, no. 6, pp. 603–605, Aug. 2020, ISSN: 1538-3598. DOI: 10.1001/JAMA.2020.12603. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32644129/.

[21] G. EL, C. JR, O. ZS, L. PH, S. AL, T. C, D. RM, J. DT, B. R, H. SU, B. A, L. EM, and K. IJ, "Persistent neurologic symptoms and cognitive dysfunction in non-hospitalized Covid-19 "long haulers
", *Annals of clinical and translational neurology*, vol. 8, no. 5, pp. 1073–1085, May 2021, ISSN: 2328-9503. DOI: 10.1002/ACN3.51350. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33755344/.

[22] H. V, S. D, D. EP, and P. I, "COVID-19: from an acute to chronic disease? Potential long-term health consequences", *Critical reviews in clinical laboratory sciences*, vol. 58, no. 5, pp. 297–310, 2021, ISSN: 1549-781X. DOI: 10.1080/10408363.2020.1860895. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33347790/.

[23] D. W. Carley and S. S. Farabi, "Physiology of Sleep", *Diabetes Spectrum : A Publication of the American Diabetes Association*, vol. 29, no. 1, p. 5, Feb. 2016. DOI: 10.2337/DIASPECT.29.1.5. [Online]. Available: /pmc/articles/PMC4755451/%20/pmc/articles/PMC4755451/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4755451/.

[24] Z. Alimoradi, A. Broström, H. W. Tsang, M. D. Griffiths, S. Haghayegh, M. M. Ohayon, C.-Y. Lin, and A. H. Pakpour, "Sleep problems during COVID-19 pandemic and its' association to psychological distress: A systematic review and meta-analysis", *EClinicalMedicine*, vol. 36, p. 100 916, Jun. 2021, ISSN: 2589-5370. DOI: 10.1016/J.ECLINM.2021.100916. [Online]. Available: http://www.thelancet.com/article/S2589537021001966/fulltext%20http://www.thelancet.com/article/S2589537021001966/abstract%20https://www.thelancet.com/journals/eclinm/article/PIIS2589-5370(21)00196-6/abstract.

[25] C. S. Nayak and A. C. Anilkumar, "EEG Normal Sleep", *StatPearls*, Jul. 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK537023/.

[26] D. O'Donnell, E. J. Silva, M. Munch, J. M. Ronda, W. Wang, and J. F. Duffy, "Comparison of Subjective and Objective Assessments of Sleep in Healthy Older Subjects Without Sleep Complaints", *Journal of sleep research*, vol. 18, no. 2, p. 254, Jun. 2009. DOI: 10.1111/J.1365-2869.2008.00719.X. [Online]. Available: /pmc/articles/PMC2975570/%20/pmc/articles/PMC2975570/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2975570/.

[27] D. DJ and L. HP, "Sleep Physiology, Circadian Rhythms, Waking Performance and the Development of Sleep-Wake Therapeutics", *Handbook of experimental pharmacology*, vol. 253, pp. 441–481, 2019, ISSN: 0171-2004. DOI: 10.1007/164{_}2019{_}243. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/31254050/.

[28] L. Xu, F. Han, B. T. Keenan, E. Kneeland-Szanto, H. Yan, X. Dong, Y. Chang, L. Zhao, X. Zhang, J. Lixs, A. I. Pack, and S. T. Kuna, "Validation of the Nox-T3 portable monitor for diagnosis of obstructive sleep apnea in Chinese adults", *Journal of Clinical Sleep Medicine*, vol. 13, no. 5, pp. 675–683, 2017, ISSN: 15509397. DOI: 10.5664/jcsm.6582.

[29] S. Mukherjee, S. R. Patel, S. N. Kales, N. T. Ayas, K. P. Strohl, D. Gozal, and A. Malhotra, "An Official American Thoracic Society Statement: The Importance of Healthy Sleep. Recommendations and Future Priorities", *https://doi.org/10.1164/rccm.201504-0767ST*, vol. 191, no. 12, pp. 1450–1458, Jun. 2015. DOI: 10.1164/RCCM.201504-0767ST. [Online]. Available: www.atsjournals.org.

[30] C. C. Panel, N. F. Watson, M. S. Badr, G. Belenky, D. L. Bliwise, O. M. Buxton, D. Buysse, D. F. Dinges, J. Gangwisch, M. A. Grandner, C. Kushida, R. K. Malhotra, J. L. Martin, S. R. Patel, S. F. Quan, and E. Tasali, "Recommended Amount of Sleep for a Healthy Adult: A Joint Consensus Statement of the American Academy of Sleep Medicine and Sleep Research Society", *Sleep*, vol. 38, no. 6, p. 843, Jun. 2015. DOI: 10.5665/SLEEP.4716. [Online]. Available: /pmc/articles/PMC4434546/%20/

pmc/articles/PMC4434546/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4434546/.

[31] M. P. Walker, *Why we sleep : unlocking the power of sleep and dreams*. 2017, p. 360, ISBN: 1501144316.

[32] M. R. Zielinski, J. T. McKenna, and R. W. McCarley, "Functions and mechanisms of sleep", *AIMS Neuroscience*, vol. 3, no. 1, pp. 67–104, 2016, ISSN: 23737972. DOI: 10.3934/NEUROSCIENCE.2016.1.67/FULLTEXT.HTML. [Online]. Available: /pmc/articles/PMC5390528/%20/pmc/articles/PMC5390528/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5390528/.

[33] C. N, C. N, M. G, and C. S, "Changes in sleep pattern, sense of time and digital media use during COVID-19 lockdown in Italy", *Journal of sleep research*, vol. 29, no. 4, Aug. 2020, ISSN: 1365-2869. DOI: 10.1111/JSR.13074. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32410272/.

[34] L. LY, W. J, O.-Y. XY, M. Q, C. R, L. FX, Z. YP, T. Q, and W. T, "The immediate impact of the 2019 novel coronavirus (COVID-19) outbreak on subjective sleep status", *Sleep medicine*, vol. 77, pp. 348–354, Jan. 2021, ISSN: 1878-5506. DOI: 10.1016/J.SLEEP.2020.05.018. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32593614/.

[35] C. H. Bastien, A. Vallières, and C. M. Morin, "Validation of the Insomnia Severity Index as an outcome measure for insomnia research", *Sleep Medicine*, vol. 2, no. 4, pp. 297–307, Jul. 2001, ISSN: 1389-9457. DOI: 10.1016/S1389-9457(00)00065-4.

[36] D. J, Z. F, H. W, S. Z, W. CY, C. O, H. E, and Z. QK, "The prevalence of depression, anxiety, and sleep disturbances in COVID-19 patients: a meta-analysis", *Annals of the New York Academy of Sciences*, vol. 1486, no. 1, pp. 90–111, Jan. 2021, ISSN: 1749-6632. DOI: 10.1111/NYAS.14506. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33009668/.

[37] C. JM, B.-R. C, K.-K. CK, N. PG, M. JN, M. SE, D. RD, G. MA, and L. PR, "Prevalence of symptoms of depression, anxiety, insomnia, posttraumatic stress disorder, and psychological distress among populations affected by the COVID-19 pandemic: A systematic review and meta-analysis", *Psychiatry research*, vol. 295, Jan. 2021, ISSN: 1872-7123. DOI: 10.1016/J.PSYCHRES.2020.113599. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33285346/.

[38] M. MJ, R. SA, and A. GM, "Sleep disturbances in patients with post-traumatic stress disorder: epidemiology, impact and approaches to management", *CNS drugs*, vol. 20, no. 7, pp. 567–590, 2006, ISSN: 1172-7047. DOI: 10.2165/00023210-200620070-00003. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/16800716/.

[39] C. M. Morin and J. Carrier, "The acute effects of the COVID-19 pandemic on insomnia and psychological symptoms", *Sleep Medicine*, vol. 77, p. 346, Jan. 2021. DOI: 10.1016/J.SLEEP.2020.06.005. [Online]. Available: /pmc/articles/PMC7274952/%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7274952/.

[40] M. Opp, L. T. .-.-. F. Biosci, and u. 2003, "Neural-immune interactions in the regulation of sleep", *researchgate.net*, 2003. DOI: 10.2741/1061. [Online]. Available: https://www.researchgate.net/profile/Mark-Opp/publication/10800706_Neural-immune_interactions_in_the_regulation_of_sleep/links/0fcfd5087167be8355000000/Neural-immune-interactions-in-the-regulation-of-sleep.pdf.

[41] E. Späth-Schwalbe, T. Lange, B. Perras, H. Lorenz Fehm, and J. Born, "INTERFERON-α ACUTELY IMPAIRS SLEEP IN HEALTHY HUMANS", *Cytokine*, vol. 12, no. 5, pp. 518–521, May 2000, ISSN: 1043-4666. DOI: 10.1006/CYTO.1999.0587.

[42] L. Besedovsky, T. Lange, and M. Haack, "The Sleep-Immune Crosstalk in Health and Disease", *Physiological Reviews*, vol. 99, no. 3, p. 1325, Jul. 2019. DOI: 10.1152/PHYSREV.00010.2018. [Online]. Available: /pmc/articles/PMC6689741/%20/pmc/articles/PMC6689741/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6689741/.

[43] E. G. Ibarra-Coronado, A. M. Pantaleón-Martínez, J. Velazquéz-Moctezuma, O. Prospéro-García, M. Méndez-Díaz, M. Pérez-Tapia, L. Pavón, and J. Morales-Montor, "The Bidirectional Relationship between Sleep and Immunity against Infections", *Journal of Immunology Research*, vol. 2015, 2015. DOI: 10.1155/2015/678164. [Online]. Available: /pmc/articles/PMC4568388/%20/pmc/articles/PMC4568388/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4568388/.

[44] S.-G. O, M. A, V. V, K. M, S. E, A. V, Y. T, Z. D, P. T, and K. J, "Brain Mechanisms of COVID-19-Sleep Disorders", *International journal of molecular sciences*, vol. 22, no. 13, Jul. 2021, ISSN: 1422-0067. DOI: 10.3390/IJMS22136917. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/34203143/.

[45] C. A. Goldstein, R. B. Berry, D. T. Kent, D. A. Kristo, A. A. Seixas, S. Redline, and M. Brandon Westover, "Artificial intelligence in sleep medicine: background and implications for clinicians", *Journal of Clinical Sleep Medicine : JCSM : Official Publication of the American Academy of Sleep Medicine*, vol. 16, no. 4, p. 609, Apr. 2020, ISSN: 15509397. DOI: 10.5664/JCSM.8388. [Online]. Available: /pmc/articles/PMC7161463/%20/pmc/articles/PMC7161463/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7161463/.

[46] N. F. Watson and C. R. Fernandez, "Artificial intelligence and sleep: Advancing sleep medicine", *Sleep medicine reviews*, vol. 59, Oct. 2021, ISSN: 1532-2955. DOI: 10.1016/J.SMRV.2021.101512. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/34166990/.

[47] U. R. Acharya, H. Fujita, V. K. Sudarshan, S. Bhat, and J. E. Koh, "Application of entropies for automated diagnosis of epilepsy using EEG signals: A review", *Knowledge-Based Systems*, vol. 88, pp. 85–96, Nov. 2015, ISSN: 0950-7051. DOI: 10.1016/J.KNOSYS.2015.08.004.

[48] X. Wang, G. Gong, N. Li, and S. Qiu, "Detection analysis of epileptic EEG using a novel random forest model combined with grid search optimization", *Frontiers in Human Neuroscience*, vol. 13, p. 52, Feb. 2019, ISSN: 16625161. DOI: 10.3389/FNHUM.2019.00052/BIBTEX.

[49] P. Chriskos, C. A. Frantzidis, C. M. Nday, P. T. Gkivogkli, P. D. Bamidis, and C. Kourtidou-Papadeli, "A review on current trends in automatic sleep staging through bio-signal recordings and future challenges", *Sleep Medicine Reviews*, vol. 55, p. 101 377, Feb. 2021, ISSN: 1087-0792. DOI: 10.1016/J.SMRV.2020.101377.

[50] C. M. Bishop, *Pattern Recognition and Machine Learning*, M. Jordan, J. Kleinberg, and B. Schölkopf, Eds. Singapore: Springer, 2006.

[51] A. Burkov, *The hundred-page machine learning book*, English. 2019, OCLC: 1089445188, ISBN: 978-1-9995795-0-0.

[52] L. Breiman, "Random Forests", vol. 45, pp. 5–32, 2001.

[53] R. Joshi, *Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog*, 2016. [Online]. Available: https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/.

[54] K. Hajian-Tilaki, "Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation", *Caspian Journal of Internal Medicine*, vol. 4, no. 2, p. 627, 2013, ISSN: 20086164. [Online]. Available: /pmc/articles/PMC3755824/%20/pmc/articles/PMC3755824/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3755824/.

[55] *ÍSLENSK ERFÐAGREINING - ÞEKKING Í ALLRA ÞÁGU*. [Online]. Available: https://www.decode.is/.

[56] *HEILSURANNSÓKN | Turninn, Smáratorgi 3, 201 Kópavogi, 4. hæð*. [Online]. Available: https://www.heilsurannsokn.is/.

[57] S. Kainulainen, H. Korkalainen, S. Sigurdardottir, S. Myllymaa, M. Serwatko, S. P. Sigurdardottir, M. Clausen, T. Leppanen, and E. S. Arnardottir, "Comparison of EEG Signal Characteristics between Polysomnography and Self Applied Somnography Setup in a Pediatric Cohort", *IEEE Access*, vol. 9, pp. 110 916–110 926, Jan. 2021, ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3099987. [Online]. Available: https://www.hirsla.lsh.is/handle/2336/621930.

[58] R. B. Berry, R. Brooks, C. E. Gamaldo, S. M. Harding, R. M. Lloyd, C. L. Marcus, and B. V. Vaughn, "AASM | Scoring Manual Version 2.5 The AASM Manual for the Scoring of Sleep and Associated Events RULES, TERMINOLOGY AND TECHNICAL SPECIFICATIONS VERSION 2.2", 2018. [Online]. Available: www.aasmnet.org..

[59] D. L. Reed and W. P. Sacco, "Measuring Sleep Efficiency : What Should the Denominator Be ?", vol. 12, no. 2, pp. 12–15, 2016.

[60] S. K. Kim, C. K. Rhee, J. Y. Kang, J. S. Kim, J. W. Kim, Y. H. Kim, S. H. Lee, S. S. Kwon, and H. S. Moon, "RELATIONSHIP BETWEEN AROUSAL INDEX AND CLINICAL MANIFESTATION IN PATIENTS WHO PERFORMED POLYSOMNOGRAPHY", *CHEST*, vol. 132, no. 4, 648B, Oct. 2007, ISSN: 0012-3692. DOI: 10.1378/CHEST.132.4{_}MEETINGABSTRACTS.648B. [Online]. Available: http://journal.chestnet.org/article/S0012369216444430/fulltext.

[61] M. A. Albqoor and A. M. Shaheen, "Sleep quality, sleep latency, and sleep duration: a national comparative study of university students in Jordan", *Sleep and Breathing*, vol. 25, no. 2, pp. 1147–1154, Jun. 2021, ISSN: 15221709. DOI: 10.1007/S11325-020-02188-W/TABLES/3. [Online]. Available: https://link.springer.com/article/10.1007/s11325-020-02188-w.

[62] D. A. Pevernagie, B. Gnidovec-Strazisar, L. Grote, R. Heinzer, W. T. McNicholas, T. Penzel, W. Randerath, S. Schiza, J. Verbraecken, and E. S. Arnardottir, "On the rise and fall of the apnea-hypopnea index: A historical review and critical appraisal", *Journal of sleep research*, vol. 29, no. 4, Aug. 2020, ISSN: 1365-2869. DOI: 10.1111/JSR.13066. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32406974/.

[63] N. H. Rashid, S. Zaghi, M. Scapuccin, M. Camacho, V. Certal, and R. Capasso, "The Value of Oxygen Desaturation Index for Diagnosing Obstructive Sleep Apnea: A Systematic Review", *The Laryngoscope*, vol. 131, no. 2, pp. 440–447, Feb. 2021, ISSN: 1531-4995. DOI: 10.1002/LARY.28663. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/lary.28663.

[64] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009, ISBN: 1441412697.

[65] R Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013. [Online]. Available: http://www.R-project.org/.

[66] R. V. Lenth, "Algorithm as 243: Cumulative distribution function of the non-central t distribution", *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 38, no. 1, pp. 185–189, 1989, ISSN: 00359254, 14679876. [Online]. Available: http://www.jstor.org/stable/2347693 (visited on 04/18/2022).

[67] *pingouin.power_ttest — pingouin 0.5.1 documentation*. [Online]. Available: https://pingouin-stats.org/generated/pingouin.power_ttest.html.

[68] M. Kay, L. A. Elkin, J. J. Higgins, and J. O. Wobbrock, *ARTool: Aligned rank transform for nonparametric factorial anovas*, R package version 0.11.1, 2021. DOI: 10.5281/zenodo.594511. [Online]. Available: https://github.com/mjskay/ARTool.

[69] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, *et al.*, "Top 10 algorithms in data mining", *Knowledge and information systems*, vol. 14, no. 1, pp. 1–37, 2008.

[70] M. T. Camacho Olmedo, M. Paegelow, J. F. Mas, and F. Escobar, "Geomatic Approaches for Modeling Land Change Scenarios. An Introduction", pp. 1–8, 2018. DOI: 10.1007/978-3-319-60801-3{_}1.

[71] R. Boostani, F. Karimzadeh, and M. Nami, "A comparative review on sleep stage classification methods in patients and healthy individuals", *Computer Methods and Programs in Biomedicine*, vol. 140, pp. 77–91, Mar. 2017, ISSN: 0169-2607. DOI: 10.1016/J.CMPB.2016.12.004.

[72] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy", *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: 10.1038/s41586-020-2649-2. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2.

[73] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python", *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[74] O. Barzilay and A. Wolf, "A fast implementation for EMG signal linear envelope computation", *Journal of electromyography and kinesiology : official journal of the International Society of Electrophysiological Kinesiology*, vol. 21, no. 4, pp. 678–682, Aug. 2011, ISSN: 1873-5711. DOI: 10.1016/J.JELEKIN.2011.04.004. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/21689608/.

[75] C. de Boor, *A Practical Guide to Splines*, ser. Applied Mathematical Sciences. New York, NY: Springer New York, 1978, vol. 27, ISBN: 978-0-387-90356-9. DOI: 10.1007/978-1-4612-6333-3.

[76] P. D. Welch, "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms", *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967, ISSN: 00189278. DOI: 10.1109/TAU.1967.1161901.

[77] J. O. Smith, "Spectral Audio Signal Processing", *Center for Computer Research in Music and Acoustics (CCRMA)*, pp. 1–674, 2011. [Online]. Available: https://ccrma.stanford.edu/~jos/sasp/%20http://books.w3k.org/.

[78] A. M. Sayeed, A. Papandreou-Suppappola, S. B. Suppappola, X. G. Xia, F. Hlawatsch, G. Matz, B. Boashash, G. Azemi, and N. A. Khan, "Detection, Classification, and Estimation in the (t,f) Domain", *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference*, pp. 693–743, Jan. 2016. DOI: 10.1016/B978-0-12-398499-9.00012-1.

[79] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[80] A. M. Musolf, E. R. Holzinger, J. D. Malley, and J. E. Bailey-Wilson, "What makes a good prediction? Feature importance and beginning to open the black box of machine learning in genetics", *Human Genetics*, pp. 1–14, Dec. 2021, ISSN: 14321203. DOI: 10.1007/S00439-021-02402-Z/FIGURES/5. [Online]. Available: https://link.springer.com/article/10.1007/s00439-021-02402-z.

[81] Y. Jung and J. Hu, "A K-fold Averaging Cross-validation Procedure", *Journal of nonparametric statistics*, vol. 27, no. 2, pp. 167–179, Apr. 2015, ISSN: 1048-5252. DOI: 10.1080/10485252.2015.1010532. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/27630515/.

[82] Y. Bengio, "Practical Recommendations for Gradient-Based Training of Deep Architectures", 2012. [Online]. Available: http://deeplearning.net/software/pylearn2.

[83] C. Cooney, A. Korik, R. Folli, and D. Coyle, "Evaluation of Hyperparameter Optimization in Machine and Deep Learning Methods for Decoding Imagined Speech EEG", *Sensors 2020, Vol. 20, Page 4629*, vol. 20, no. 16, p. 4629, Aug. 2020, ISSN: 1424-8220. DOI: 10.3390/S20164629. [Online]. Available: https://www.mdpi.com/1424-8220/20/16/4629/htm%20https://www.mdpi.com/1424-8220/20/16/4629.

[84] S. Sharma, S. Sharma, and A. Athaiya, "ACTIVATION FUNCTIONS IN NEURAL NETWORKS", *International Journal of Engineering Applied Sciences and Technology*, vol. 4, pp. 310–316, 2020. [Online]. Available: http://www.ijeast.com.

[85] D. Helgason, I. Hinriksdottir, and R. Ingvarsson, "Symptoms, physical measures and cognitive tests after SARS-CoV-2 infection in a large population-based case-control study", Apr. 2022. DOI: 10.21203/RS.3.RS-1539784/V1. [Online]. Available: https://www.researchsquare.com%20https://www.researchsquare.com/article/rs-1539784/v1.

[86] *Symptoms of COVID-19 | CDC*. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html.

[87] U. Orhan, M. Hekim, and M. Ozer, "EEG signals classification using the K-means clustering and a multilayer perceptron neural network model", *Expert Systems with Applications*, vol. 38, no. 10, pp. 13 475–13 481, Sep. 2011, ISSN: 09574174. DOI: 10.1016/J.ESWA.2011.04.149.

Appendix A

Best Hyperparameters of Models

Table A.1: The best performing hyper-parameters used for the models trained on the whole dataset. Parameter:**Value**

Decision Tree	**Random Forest**	**MLP**
criterion : **'gini'**	n_estimators : **100**	hidden_layer_size : **(50, 100, 100)**
max_depth : **13**	criterion : **'gini'**	activation : **'tanh'**
min_samples_split : **30**	max_depth : **100**	solver : **'sgd'**
max_features : **'log2'**	min_samples_split : **40**	max_iter : **1500**
min_samples_leaf : **4**	max_features : **'log2'**	learning_rate : **'adaptive'**
	min_samples_leaf : **10**	learning_rate_init : **0.01**
	bootstrap : **True**	

Table A.2: The best performing hyper-parameters used for the models trained on the N1 dataset. Parameter:**Value**

Decision Tree	**Random Forest**	**MLP**
criterion : **'entropy'**	n_estimators : **100**	hidden_layer_size : **(50, 100, 100)**
max_depth : **12**	criterion : **'entropy'**	activation : **'relu'**
min_samples_split : **40**	max_depth : **100**	solver:**'sgd'**
max_features : **'sqrt'**	min_samples_split : **30**	max_iter : **2500**
min_samples_leaf : **6**	max_features : **'sqrt'**	learning_rate : **'constant'**
	min_samples_leaf : **1**	learning_rate_init : **0.1**
	bootstrap : **False**	

Table A.3: The best performing hyper-parameters used for the models trained on the N2 dataset. Parameter:**Value**

Decision Tree	**Random Forest**	**MLP**
criterion : **'gini'**	n_estimators : **200**	hidden_layer_size : **(100, 100)**
max_depth : **9**	criterion : **'gini'**	activation : **'relu'**
min_samples_split : **50**	max_depth : **150**	solver : **'sgd'**
max_features : **'log2'**	min_samples_split : **50**	max_iter : **1500**
min_samples_leaf : **4**	max_features : **'sqrt'**	learning_rate : **'adaptive'**
	min_samples_leaf : **4**	learning_rate_init : **0.1**
	bootstrap:**True**	

Table A.4: The best performing hyper-parameters used for the models trained on the N3 dataset. Parameter:**Value**

Decision Tree	**Random Forest**	**MLP**
criterion : **'entropy'**	n_estimators : **100**	hidden_layer_size : **(100, 100, 100)**
max_depth : **11**	criterion : **'gini'**	activation : **'tanh'**
min_samples_split:**50**	max_depth : **50**	solver : **'sgd'**
max_features : **'log2'**	min_samples_split : **50**	max_iter : **1500**
min_samples_leaf : **1**	max_features : **'log2'**	learning_rate : **'adaptive'**
	min_samples_leaf : **4**	learning_rate_init : **0.01**
	bootstrap : **False**	

Table A.5: The best performing hyper-parameters used for the models trained on the REM dataset. Parameter:**Value**

Decision Tree	**Random Forest**	**MLP**
criterion : **'entropy'**	n_estimators : **100**	hidden_layer_size : **(50, 100)**
max_depth : **14**	criterion : **'gini'**	activation : **'relu'**
min_samples_split : **50**	max_depth : **50**	solver : **'sgd'**
max_features : **'log2'**	min_samples_split : **20**	max_iter : **2500**
min_samples_leaf : **4**	max_features : **'log2'**	learning_rate : **'adaptive'**
	min_samples_leaf : **1**	learning_rate_init : **0.1**
	bootstrap : **True**	

Table A.6: The best performing hyper-parameters used for the models trained on the wake dataset. Parameter:**Value**

Decision Tree	**Random Forest**	**MLP**
criterion : **'entropy'**	n_estimators : **200**	hidden_layer_size : **(100, 100, 100)**
max_depth : **12**	criterion : **'entropy'**	activation : **'relu'**
min_samples_split : **30**	max_depth : **150**	solver : **'sgd'**
max_features : **'log2'**	min_samples_split : **20**	max_iter : **1500**
min_samples_leaf : **4**	max_features : **'log2'**	learning_rate : **'adaptive'**
	min_samples_leaf : **4**	learning_rate_init : **0.01**
	bootstrap : **False**	

Appendix B

Code

Listing B.1: Function that reads an .edf file and returns signals signal headers and recording header.

```
def read_edf_file(path):
    #Function takes in path to .edf file and returns raw signals, signal headers and headers
    signals, signal_headers, header = pyedflib.highlevel.read_edf(path)

    return signals, signal_headers, header
```

Listing B.2: Function for getting timestamps for start and end of night from scoring analysis of the study.

```
def get_timestamps(path):
    #Function takes in path to excel scoring files directory(filenames need to be the participants identifier: 'ID.xls')
    #Returns start and end time of night for each study scoring in directory as a dataframe with participant ID as index
    os.chdir(path)

    start_times = []
    end_times = []
    ID = []
    for resi_files in os.listdir(path):
         if os.path.isfile(resi_files):
                    ID.append(str(resi_files.rstrip('_Svefnskorun.xls')))
                    myworkbook=xlrd.open_workbook(resi_files)
                    ws= myworkbook.sheet_by_index(0)
                    start_times.append(xlrd.xldate_as_datetime(ws.cell_value(rowx = 3, colx = 1), 0))
                    last_row = ws.nrows -1
                    end_times.append(xlrd.xldate_as_datetime(ws.cell_value(rowx = last_row, colx = 2), 0))
    time_df = pd.DataFrame({'Start' : start_times,
                        'End' : end_times},
                       index = ID)
    return time_df
```

Listing B.3: Function for checkin for gaps in scoring in the scoring file.

```
def get_gaps(score_frame):
```

```
    #Function takes in path to csv scoring files directory(filenames need to be the participants identifier: 'ID.csv')

    #Initialize lists
    gap_start = []
    gap_end = []
    datetime_list = []
    list_to_search = []
    gap_ind = []
    diff_list = []
    dur_list = []

    #Trim to only get the list of starttimes of scored epochs
    lst = score_frame.drop([0, 1])
    lst = lst.iloc[:,1]
    lst = lst.reset_index()
    lst = lst.drop(columns=['index'])

    #Substring of just the time in the format hh:mm:ss
    for i in range(len(lst)):
        list_to_search.append(lst['Start Time'][i][0:8])

   #Change to datetime format
    for i in range(len(list_to_search)):
        datetime_list.append(datetime.strptime(list_to_search[i], '%H:%M:%S'))

    for i in range(len(datetime_list) - 1):
        #Calculate the difference in time between two subsequent scorings
        diff = datetime_list[i + 1] - datetime_list[i]

        #Check if gap between epochs is more than 30 sek
        if diff.total_seconds() > 30:
            gap_ind.append(i + 1)
            gap_start.append(datetime_list[i] + timedelta(seconds=30))
            gap_end.append(datetime_list[i + 1])
            diff_list.append(datetime_list[i + 1] - (datetime_list[i] + timedelta(seconds=30)))
            dur_list.append((datetime_list[i + 1] - (datetime_list[i] + timedelta(seconds=30))) / timedelta(seconds=30))


    #Make datafram with start of gap, end of gap, index of first epoch in gap and duration of gap in epochs
    gaps = pd.DataFrame({'Start' : gap_start,
                         'End' : gap_end,
                         'Ind' : gap_ind,
                         'Diff' : diff_list,
                         'Dur' : dur_list})



    return gaps
```

Listing B.4: Function for that takes in epochized signal and information of gaps in recording and returns the epochized signal where the gaps have been removed.

```
def remove_gaps(epochized_signal, gaps):

```

```
    #Loop throug all gaps in the recording and remove them by using their indexes
    only_sleep = epochized_signal.copy()
    for i in range(len(gaps)):
        start_index = int(gaps['Ind'][i])
        end_index = int(gaps['Ind'][i] + gaps['Dur'][i])
        only_sleep = only_sleep.drop(only_sleep.index[start_index:end_index])

    return only_sleep
```

Listing B.5: Function for extracting and preparing the EEG singal for further processing.

```
def prep_EEG_signal(Signals, Signal_Headers, Header, Start_of_night, End_of_night
    , Fs):

    #This function takes in the raw signal, the channel names and the recording
    start and end analysis times and sampling frequency
    #Prepares the signal by identifying the EEG channels and adding them to a
    list and cuts out irrelevant data from each channel(data that was not part
    of the night and therefore not scored)

    print('Prepping signal...')

    #Initialize lists and counter i
    channel_names = []
    EEG_sleep = []
    EEGsignal_list = []
    i = 0
    indexes = []

    #Make list with only channel names
    for signal in Signal_Headers:
        channel_names.append(signal['label'])

    #Search for the channels to be used and add their indexes to a list
    for names in channel_names:
        if names == 'AF3-E3E4':
            indexes.append(i)
        elif names == 'AF7-E3E4':
            indexes.append(i)
        elif names == 'AF8-E3E4':
            indexes.append(i)
        elif names == 'AF4-E3E4':
            indexes.append(i)
        i=i+1
    #Select only the channels to be used and put into a list
    for i in indexes:
        EEGsignal_list.append(Signals[i])

    #Identify the indexes where scored recording starts and ends
    start_index = int(((Start_of_night - Header['startdate']).total_seconds()) *
    Fs)
    end_index = int(((End_of_night - Header['startdate']).total_seconds()) * Fs)

    #make new list with only scored recording
    for i in EEGsignal_list:
        EEG_sleep.append(i[start_index:end_index])

    #Add the names of the channels to be used to a list
```

```
    k = 0
    channel_use = []
    for i in indexes:
        channel_use.append(Signal_Headers[i]['label'])
        k = k + 1

    #Return list of scored signals of the desired channels and the names of the channels used
    return EEG_sleep, channel_use
```

Listing B.6: Function that splits the raw signal into epochs.

```
def epochize(data, channel_names, epoch_len, sampling_freq, start_timestamp):
    #Function that takes in the raw prepped signal, preferred epoch length, sampling frequency of the signal and start time of signal
    #Returns dataframe with epochized signal(channel name as column header for epochs) and timestamp of each epoch start

    print('Starting epochization...')

    #Find the lengt of the signal
    l = len(data)

    #Find the number of data points
    number = int(epoch_len * sampling_freq)

    #initialize the epoch list
    epochs = []
    num_epochs = int(np.ceil(l / number))
    #add the epochs to the list
    for x in range(0, l, number):

        epochs.append(data[x:x+number])

    #initialize a list of timestamps at the beginning of each epoch
    timestamps_epoch_start = []

    for i in range(0, num_epochs):

        #make start timestamps for all the subsequent epochs and add to list
        next_timestamp = start_timestamp + datetime.timedelta(seconds=epoch_len * i)

        timestamps_epoch_start.append(next_timestamp)

    # make dataframe with timestamps and epoch
    df = pd.DataFrame()
    df['Epoch start'] = timestamps_epoch_start
    df[channel_names] = epochs

    return df
```

Listing B.7: Function that makes the upper and lower signal envelopes for an epoch.

```
def upper_lower_signal_envelopes(epoch):
    #Takes in an epoch
```

```
    #Calculates and returns the upper and lower signal envelopes using cubic spline
    #Code from author A_A on StackOverflow (https://stackoverflow.com/questions/34235530/how-to-get-high-and-low-envelope-of-a-signal)

    upper = zeros(epoch.shape)
    q_l = zeros(epoch.shape)

    #Define starting points for upper and lower envelopes as x and y coordinates.
    upper_x = [0,]
    upper_y = [epoch[0],]

    lower_x = [0,]
    lower_y = [epoch[0],]

    #Detect peaks and valleys and add to coordinate lists.
    for k in range(1,len(epoch)-1):
        if (sign(epoch[k]-epoch[k-1])==1) and (sign(epoch[k]-epoch[k+1])==1):
            upper_x.append(k)
            upper_y.append(epoch[k])

        if (sign(epoch[k]-epoch[k-1])==-1) and ((sign(epoch[k]-epoch[k+1]))==-1):
            lower_x.append(k)
            lower_y.append(epoch[k])

    #Define the end point as the end of the epoch an append to coordinate lists
    upper_x.append(len(epoch)-1)
    upper_y.append(epoch[-1])

    lower_x.append(len(epoch)-1)
    lower_y.append(epoch[-1])

    #Interpolate using the set of upper peak coordinates and lower valley coordinates using cubic spline.
    upper_i = CubicSpline(upper_x,upper_y)
    lower_i = CubicSpline(lower_x,lower_y)

    #Make envelope values
    for i in range(0,len(epoch)):
        envelope_upper[i] = upper_i(i)
        envelope_lower[i] = lower_i(i)
    return envelope_upper, envelope_lower
```

Listing B.8: Function that calculates the Welch PSD

```
def welch_PSD(epoch, fs, filter_order):
    #Function that calculates power spectra of epoch with Welch's method - takes in epoch, sampling frequency and filter order
    #Returns frequencies(F), power spectral density(PSD) and absolute power using Simpson's rule to estimate the area under the curve

    window_size = int(fs * filter_order)

    F, PSD = signal.welch(epoch, fs, nperseg=window_size)

    # Frequency resolution
    freq_res = F[1] - F[0]  # = 1 / 4 = 0.25

```

```
    # Compute the absolute power by approximating the area under the curve with ↩
    ↪simpsons rule
    abs_power = simps(PSD, dx=freq_res)

    return F, PSD, abs_power
```

Listing B.9: Function that calculates the entropy from PSD.

```
def s_entropy(PSD):
    #Function takes in power spectral density
    #Returns spectral entropy

    normPSD = PSD / PSD.sum(axis=-1, keepdims=True)

    spectral_entropy = -(normPSD * np.log2(normPSD)).sum(axis=-1)

    return spectral_entropy
```

Listing B.10: Function that finds features for all epochs of an epochized signal.

```
def find_features(epochized_signal, sampling_freq):
    #Function that finds epochwise features - takes in one datafram of epochized ↩
    ↪signal and the sampling frequency of the signal
    #Returns dataframe with features of each epoch
    print('Starting featurization...')

    #find how many epochs there are in the signal
    l = len(epochized_signal)

    #initialize all lists
    dataframe = pd.DataFrame()
    mini = []
    maxi = []
    med = []
    var = []
    std = []
    cov = []
    skew_list = []
    kurt_list = []
    upper_mean = []
    lower_mean = []
    upper_std = []
    lower_std = []
    upper_var = []
    lower_var = []
    upper_skew = []
    lower_skew = []
    upper_kurt = []
    lower_kurt = []
    PSD = []
    abs_power = []
    entropy = []

    #loop through epochs and calulate the features and add to relevant feature ↩
    ↪list
    for i in range(0, l):
```

```
        mini.append(np.min(np.abs(epochized_signal['n'][i])))
        maxi.append(np.max(np.abs(epochized_signal['n'][i])))
        med.append(np.median(epochized_signal['n'][i]))
        std.append(np.std(epochized_signal['n'][i]))
        var.append(np.var(epochized_signal['n'][i]))
        skew_list.append(skew(epochized_signal['n'][i]))
        kurt_list.append(kurtosis(epochized_signal['n'][i]))

        upper, lower = upper_lower_signal_envelopes(epochized_signal['n'][i])
        upper_mean.append(np.mean(upper))
        lower_mean.append(np.mean(lower))
        upper_std.append(np.std(upper))
        lower_std.append(np.std(lower))
        upper_var.append(np.var(upper))
        lower_var.append(np.var(lower))
        upper_skew.append(skew(upper))
        lower_skew.append(skew(lower))
        upper_kurt.append(kurtosis(upper))
        lower_kurt.append(kurtosis(lower))


        _, PSDlist, power = welch_PSD(epochized_signal['n'][i], sampling_freq, 4)
        abs_power.append(power)
        entropy.append(s_entropy(PSDlist))
        PSD.append(PSDlist)

    #add the epochs to the feature dataframe
    dataframe = epochized_signal

    #add all feature list as columns to the feature dataframe
    dataframe['Epoch min'] = mini
    dataframe['Epoch max'] = maxi
    dataframe['Epoch median'] = med
    dataframe['Epoch var'] = var
    dataframe['Epoch std'] = std
    dataframe['Skewness'] = skew_list
    dataframe['Kurtosis'] = kurt_list
    dataframe['Upper env mean'] = upper_mean
    dataframe['Lower env mean'] = lower_mean
    dataframe['Upper env std'] = upper_std
    dataframe['Lower env std'] = lower_std
    dataframe['Upper env var'] = upper_var
    dataframe['Lower env var'] = lower_var
    dataframe['Upper env skew'] = upper_skew
    dataframe['Lower env skew'] = lower_skew
    dataframe['Upper env kurtosis'] = upper_kurt
    dataframe['Lower env kurtosis'] = lower_kurt
    dataframe['Abs Power'] = abs_power
    dataframe['Entropy'] = entropy


    return dataframe
```

Listing B.11: Feature extraction pipeline. Function that works from the directory where the recordings are and makes the dataset with labelled feature vectors for all epochs of all recordings.

```
def make_feature_file(edf_path, scoring_path1, scoring_path2, epoch_len, fs, time_df1, time_df2):
    #Function that takes in the path to the edf file directory, path to scoring files of first and second measurement, epoch length, sampling frequency, time dataframes for both recordings
    #Writes all epochs of all signal channels for all participant recordings in cohort into a .csv file and each epoch i labelled with the sleep stage the epoch belongs to and whether the epoch is from a measurement before or after Covid-19 infection

#-----------------------------------------------
    #Explanation of data structure and directories

    #in edf_path:
        #'ID1_1.edf'
        #'ID1_2.edf'
        #'ID2_1.edf'
        #'ID2_2.edf'
        #...
    #in scoring_path1:
        #'ID1_Svefnskorun_1.edf'
        #'ID2_Svefnskorun_1.edf'
        #...

    #in scoring_path2:
        #'ID1_Svefnskorun_2.edf'
        #'ID2_Svefnskorun_2.edf'
        #...
#-----------------------------------------------
    #set counter and initialize dataframe
    i = 0
    supreme_df = pd.DataFrame()
    #set directory
    os.chdir(edf_path)

    #Loop thorugh recordings
    for edf_files in os.listdir(edf_path):
        #make sure only the edf files are read
        if edf_files.find("checkpoint") == -1:
            signals, signal_headers, header = read_edf_file(edf_files)
            print("File successfully loaded: ", edf_files, "!!!!")
            #Check if the number 1 is not in the name of the file to distinguish between what time dataframe to get timestamps from and what scoring file to open
            if (edf_files.find("1") == -1):
                [start_time, end_time] = [time_df2.loc[edf_files.rstrip('_2.edf')]['Start'], time_df2.loc[edf_files.rstrip('_2.edf')]['End']]
                scoring = pd.read_csv(scoring_path2 + edf_files.rstrip('_2.edf') + '_Svefnskorun_2.csv', sep = ';')
                print('Length of scoring list:', len(scoring))
                #Check if the ID
                bool_C19 = True
            else:
                [start_time, end_time] = [time_df1.loc[edf_files.rstrip('_1.edf')]['Start'], time_df1.loc[edf_files.rstrip('_1.edf')]['End']]
                scoring = pd.read_csv(scoring_path2 + edf_files.rstrip('_1.edf') + '_Svefnskorun_1.csv', sep = ';')
                print('Length of scoring list:', len(scoring))
                bool_C19 = False
```

```
48
            print("Start time for this recording is: ", start_time)
50          print("End time for this recording is: ", end_time)
            print('Covid? ',bool_C19)
52
            #Prepare the signal
54          EEG_sleep, channel_names = prep_signal(signals, signal_headers, ↩
    ↪header, start_time, end_time, fs)
            print('Signal prep successful')
56
            #Epochize each channel of EEG signal
58          gaps = get_gaps(scoring)

60          df_AF3 = remove_gaps(epochize(recording_df['AF3'].to_numpy(), "n", ↩
    ↪epoch_len, fs, start_time), gaps)
            df_AF7 = remove_gaps(epochize(recording_df['AF7'].to_numpy(), "n", ↩
    ↪epoch_len, fs, start_time), gaps)
62          df_AF8 = remove_gaps(epochize(recording_df['AF8'].to_numpy(), "n", ↩
    ↪epoch_len, fs, start_time), gaps)
            df_AF4 = remove_gaps(epochize(recording_df['AF4'].to_numpy(), "n", ↩
    ↪epoch_len, fs, start_time), gaps)
64
            print('Epochization successful')
66          print('Annotate data...')
            scoring_labels = scoring.iloc[2:,0]
68          df_AF3['Scoring'] = scoring_labels
            df_AF7['Scoring'] = scoring_labels
70          df_AF8['Scoring'] = scoring_labels
            df_AF4['Scoring'] = scoring_labels
72
            print('Annotate data...')
74          #Label the epochs from second measurement as class 1(after Covid-19) ↩
    ↪and epochs from the first measurement as class 0(before Covid-19).
            if bool_C19:
76              l = len(df_AF3)
                covid = [1] * l
78              df_AF3['C19'] = covid
                df_AF7['C19'] = covid
80              df_AF8['C19'] = covid
                df_AF4['C19'] = covid
82              bool_C19 = False
            else:
84              l = len(df_AF3)
                no_covid = [0] * l
86              df_AF3['C19'] = no_covid
                df_AF7['C19'] = no_covid
88              df_AF8['C19'] = no_covid
                df_AF4['C19'] = no_covid
90          print('Annotation of data complete')

92          #Combine all channel epochs into one dataframe
            dataframe_all_epochs = pd.concat([df_AF3, df_AF7, df_AF8, df_AF4], ↩
    ↪ignore_index = True)
94
            #Find features for all epochs in the dataframe
96          feature_df = find_features(dataframe_all_epochs, fs)
            print('Features extracted')
98
            #Write the dataframe to data.csv
```

```
            #If it is the first recording then the header=True else header=Fales
            if i == 0:
                with open('data.csv', 'a') as f:
                    feature_df.to_csv(f, header=True)
            else:
                with open('data.csv', 'a') as f:
                    feature_df.to_csv(f, header=False)

            print('↩
    ↪----------------------------------------------------------------------------------------↩
    ↪')
            #Add to recording counter
            i = i + 1


    print('!!!DONE!!!')
```

Listing B.12: Function that makes a dataset based on sleep stage. Takes in the whole dataset and what sleep stage dataset we want.

```
def make_dataset(DF, stage):
    #Takes in whole dataset and sleep stage: 'N1', 'N2', 'N3', 'REM' or 'Wake'.
    #Returns a dataset of only epoch features from the desired stage.
    dataset = DF.loc[DF['Scoring'] == stage].iloc[:,1:]

    return dataset
```

Listing B.13: Function that splits a dataset into test and training sets.

```
def split(dataset, test_size, whole_dataset = False):
    #If the whole dataset is being split, both the scoring and labels need to be ↩
    ↪removed from the feature set
    if whole_dataset == False:
        X=dataset[dataset.iloc[:,1:].columns.tolist()] # Features
        y=dataset['C19']  # Labels
    else:
        X=dataset[dataset.iloc[:,2:].columns.tolist()] # Features
        y=dataset['C19']  # Labels

    # Split into training and test sets, random_state to always use the same ↩
    ↪randomization
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size↩
    ↪, random_state = 42) # 80% training and 20% test

    return X_train, X_test, y_train, y_test
```

Listing B.14: Function that performs the grid search for the Decision Tree classifier and returns the best estimator, best parameters and permutation feature importances.

```
def decision_tree_gridsearch(X_train, y_train, X_test, y_test):
    #Define parameter grid
    param_grid = {'criterion':['gini','entropy'],

                'max_depth': np.arange(3, 15),
```

```
                'max_features': ['sqrt', 'log2'],

                'min_samples_split': [20, 30, 40, 50],

                'min_samples_leaf': [1, 4, 6, 10]}
    #Grid search defined with 5-fold cross validation and maximizing the f1_macro score
    dt_gridsearch = GridSearchCV(DecisionTreeClassifier(), param_grid, cv = 5, scoring='f1_macro')

    dt_gridsearch.fit(X_train, y_train)
    #Computing feature importances with permuation
    important_features = permutation_importance(dt_gridsearch.best_estimator_, X_test, y_test,
                                n_repeats=30,
                                random_state=0)

    return dt_gridsearch.best_estimator_, dt_gridsearch.best_params_, important_features
```

Listing B.15: Function that performs the grid search for the Random Forest classifier and returns the best estimator, best parameters and permutation feature importances.

```
def random_forest_gridsearch(X_train, y_train, X_test, y_test):

    rand_sample = X_train.copy()
    rand_sample['y_train'] = y_train
    rand_sample = rand_sample.sample(1000, random_state=42)
    X = rand_sample.iloc[:,:-1]
    y = np.ravel(rand_sample.iloc[:,-1:])

    param_grid = {'n_estimators': np.arange(100, 300, 100),

    'criterion':['gini','entropy'],

    'max_features': ['sqrt','log2'],

    'max_depth': np.arange(50, 200, 50),

    'min_samples_split': [20, 30, 40, 50],

    'min_samples_leaf': [1, 4, 6, 10],

    'bootstrap': [True, False]}


    rf_gridsearch = GridSearchCV(RandomForestClassifier(), param_grid, cv = 5, scoring='f1_macro')

    rf_gridsearch.fit(X, y)

    important_features = permutation_importance(rf_gridsearch.best_estimator_, X_test, y_test,
                                n_repeats=30,
                                random_state=0)

    return rf_gridsearch.best_estimator_, rf_gridsearch.best_params_, important_features
```

Listing B.16: Function that performs the grid search for the MLP classifier and returns the best estimator, best parameters and permutation feature importances.

```
def MLP_gridsearch(X_train, y_train, X_test, y_test):

    rand_sample = X_train.copy()
    rand_sample['y_train'] = y_train
    rand_sample = rand_sample.sample(1000, random_state=42)
    X = rand_sample.iloc[:,:-1]
    y = np.ravel(rand_sample.iloc[:,-1:])

    param_grid = {'solver': ['sgd', 'adam'],

                  'activation': ['logistic', 'tanh', 'relu'],

                  'max_iter': [1500, 2500],

                  'learning_rate' : ['adaptive', 'constant'],

                  'hidden_layer_sizes': [10, 50, 100],

                  'learning_rate_init' : [0.01, 0.1]}

    cols = X.columns
    scale = StandardScaler()
    X_gs = scale.fit_transform(X)
    X_gs = pd.DataFrame(X_gs, columns=cols)

    mlp_gridsearch = GridSearchCV(MLPClassifier(), param_grid, scoring='f1_macro'↩
    ↪)

    mlp_gridsearch.fit(X_gs, y)

    important_features = permutation_importance(mlp_gridsearch.best_estimator_, ↩
    ↪X_test, y_test,
                                n_repeats=30,
                                random_state=0)

    return mlp_gridsearch.best_estimator_, mlp_gridsearch.best_params_, ↩
    ↪important_features
```

Listing B.17: Function that fits a model with best parameters to the training set and predicts and calculates, prints and returns the performance metrics of the estimator

```
def model_fit(best_params, X_train, y_train, X_test, y_test, what_model):
    #Fits the correct model to the training data(DT=Decision Tree, RF=Random ↩
    ↪Forest, MLP=Multilayer Perceptron)
    if what_model == 'DT':
        clf = DecisionTreeClassifier()
        clf.set_params(**best_params)
        clf.fit(X_train, y_train)
    elif what_model == 'RF':
        clf = RandomForestClassifier()
        clf.set_params(**best_params)
        clf.fit(X_train, y_train)
```

```
    elif what_model == 'MLP':
        clf = MLPClassifier()
        clf.set_params(**best_params)
        clf.fit(X_train, y_train)
    #Predict
    y_pred = clf.predict(X_test)
    #Claculate and preint performance metrics
    accuracy = metrics.accuracy_score(y_test, y_pred)
    recall = metrics.recall_score(y_test, y_pred)
    precision = metrics.precision_score(y_test, y_pred)
    f1 = metrics.f1_score(y_test, y_pred)
    confusion_matrix = metrics.confusion_matrix(y_test, y_pred)
    auc = metrics.roc_auc_score(y_test, y_pred)
    print('Classifier: Decision Tree')
    print("Accuracy:", accuracy)
    print('Recall: ', recall)
    print('Precision:', precision)
    print("F1 score:",f1)
    print('Confusion matrix: ')
    print(confusion_matrix)
    print('AUC: ', auc)
    return [accuracy, recall, precision, f1, confusion_matrix, auc]
```