DTU Compute
Department of Applied Mathematics and Computer Science

A critical review of Machine Learning for removing EEG artifacts

Master Thesis

A critical review of Machine Learning for removing EEG artifacts

Master Thesis
March, 2021

By
David Enslev Nyrnberg

Cover photo: Vibeke Hempler, 2012
Published by: DTU, Department of Applied Mathematics and Computer Science, Richard Petersens Plads, Building 321, 2800 Kgs. Lyngby Denmark
www.compute.dtu.dk

Abstract

Context Interest in applying complex models to EEG signals have increased the last decade. After the release of the NEDC TUH EEG Artifact Corpus in 2018 have the machine learning community taken interest in the topic. An attempt to provide a benchmark to the corpus was provided in 2019 by S. Roy, which have not been highly challenged yet.

Objective This work attempt to reproduce the benchline of S. Roy, in hope of enabling the benchline to a broader part of the EEG community. In case of reproducibility issue, a critical review of the work was applied. caper bad sentence help please

Methods The method of the benchline was applied, with edits where interpretation was needed. Edits can be observed throughout the pipeline, which was public shared on a github repository and have been used to introduce DTU-students to the same methodology.

Results The entire corpus could be loaded with these edits. A classical machine learning training, validation, testing schemed was applied on seven popular SciKit-learn models to classify six classes, five being artifacts and one null/background. The best performing model was found to be a gaussian naïve bayes model, which reached a weighted F1-score of 73.1%. IN contrast, the best performing model of the benchline was Linear Discriminant Analysis with a reported weighted F1-score of 80.12%.

Significance The thesis attempted to reproduce the benchline of S. Roy. It proved difficult due to opaque reasoning of EEG analysis, data augmentation and model application throughout the benchline. Not having public code to enable transparency resulted in the benchmark could not be reproduced. Critical points have been raised regarding the utility of the benchmark.

Contents

1 Introduction

80% of the worlds' epilepsy patients live in low middle income countries (LMICs) and even though medicine is cheap most patients remains untreated due to high diagnosis cost. BrainCapture is developing a phone based electroencephalography (EEG) capture system, suitable for LMICs. The EEG signals are then sent to experts in high income countries (HICs) for diagnosis. By enabling cheap diagnosis to LMICs a large proportion of epilepsy patients could get the treatment they need. But challenges still exist since LMICs often do not have professionals with the expert knowledge to validate the captured EEG signals before relaying them to medical experts in HICs, therefore risking that all recorded data must be discarded and re-measured.

The recently released NEDC TUH EEG Artifact Corpus (TUAR) could be of assistance to the EEG processing community in improving tools to lowering the cost of utilizing EEG for these LMICs. The corpus received a benchmark by S. Roy in 2019, which has not been challenged as of this thesis. The cornerstone of science falls on its reproducibility Plesser 2018, thus a benchmark is only worth as much as other scientists ability to utilize it though reproducibility.

1.1 Scientific pursuit

To further increase the utilization of EEG, it was the ambition of this thesis to ease the use of the TUAR corpus, partially by attempting to reproduce the work of S. Royand criticize areas of concern. Where issues with reproducing the benchline arose, the criticism aims at improving the general reproducibility of the benchline and thus make the benchline more accessible to other scientists.

A reproduction of the benchline includes: a theory section covering basic EEG, a detailed interpretation of the benchline method, experiments to explore the benchline models utilized and lastly a discussion culminating in a series of critiques when a separate party attempts to reproduce the benchline.

1.2 State of the Art

Considering the challenges of EEG, a substantial part of the community is working on automating and improving the method's efficiency, as it is a critical tool for many domains. Brain computer interfaces, online EEG methods, as well as cognitive and affective monitoring all heavily depend on the possibility of automatic EEG analysis. Treatment of medical conditions such as epilepsy, narcolepsy, or other neurological disorders/conditions depend on methods with high specificity and sensitivity¬ for making affordable diagnostics. Two reviews released in 2019 by Y. Roy et al. and Jiang, Bian, and Tian cover EEG-analysis with DL methods and Artifact removal in EEG, respectively. Both showcase the past decade's broad improvements and the general direction for EEG. Y. Roy et al. published an extensive review of EEG analysis by deep learning tools, in which they inspected all literature covering the topic. Y. Roy et al. was published in August 2019, it analysed 154 papers published in the period January 2010 to July 2018. The review of Jiang, Bian, and Tian, was published in February 2019, processing around 120 studies covering a five year period leading up to the review produced in 2018.

Y. Roy et al. highlights DL usage in EEG, focusing on objectively shown application domains, trends and interesting approaches seen in the field. The work was followed by

community guidelines in the hopes of lowering the barriers of entry and improving communication. They accomplished this by methodically querying the known study archives: PubMed, Google Scholar and arXiv for a set list of topic relevant key-terms that searched study titles and abstracts. Two of the main authors analysed the found studies and sorted them. Both peer reviewed and self-published studies were included to counteract biases introduced by a peer-review process, thus improving the study's diversity.

Y. Roy et al. showcases several of the new EEG-DL papers, published in the past decade, that seek to handle specific EEG challenges. The challenges range from domain problems, resource availability and communication strategies to novel approaches. Domain problems cover the inherent challenges the EEG method introduces. Interesting physiological and activity-specific signals tend to be drowned out by environment and artifact signals resulting in the low signal-to-noise ratio in EEG. The non-stationarity of the signal, with varying statistics across intra- and inter-patients, proves it difficult to produce classifiers. It limits the real-life application when models are rarely applied generally across datasets/patients. Access to large public datasets further emphasises the challenge. Similar challenges have shown positive results in other fields, with solutions that are emerging in several of the EEG-DL papers. Computer vision has greatly used DL methods to interpret images and time-series that have enabled computer vision tasks for real-life application. Information retrieval (by Natural language processing) has for several years improved automatic report generation by applying DL methods to lower the need for expert information [Goldberg, Driedger, and Kittredge 1994]. Anomaly detection is successfully emerging in detecting bank-fraud, analysing big data from IoT (Internet of things) or inspecting video surveillance with DL, which deals with several problems comparable to those of EEG, such as artifact (anomaly) classification, sparce data search and image processing Chalapathy and Chawla 2019.

Epilepsy, sleep, brain computer interfacing, and cognitive and affective monitoring were reported as the most frequent domains covered when applying DL for EEG analysis. Historically, EEG studies have had high variance between data amounts across studies, varying from minutes-long data sets to several thousand hours, equalling less than a hundred samples versus more than a million, depending on data usage and availability. Data accessibility is a well-known challenge in the community. It has been hard to address due to privacy concerns regarding the data from patients or subjects. These concerns often put researchers in a position that restricts how they can publish raw-, statistic- and model-data. Y. Roy et al. reports how an increasing number of studies are utilizing public data as well a clear shift from intra-subject to inter-subject.

The review points to an increasing interest for DL in EEG within the community. As EEG analysis challenges still limit EEG as a fully automated application method, several studies focus on applications that can handle these challenges. The review, thus, found six major trends seen within the field:

1. Brain-computer interfacing, sleep, epilepsy, cognitive and affective monitoring are the main domains in which DL is used for classifying EEG.

2. There is a substantial variation in the quantity of the used data, such that they range from including 1 to more than 16,000 subjects (mean = 223; median = 13), from which 62 to 9,750,000 examples (mean = 251,532; median = 14,000) and 2 to 4,800,000 minutes of EEG recording (mean = 62,602; median = 360) were produced.

3. Successful DL architectures for EEG data are focusing on CNNs, followed by RNNs

and AEs.

4. Using raw EEG as input in contrast to handcrafted features is of growing interest.
5. Compared to other baselines and benchmarks, nearly all studies could report a small improvement from using DL (median = 5.4%).
6. There is a problem of reproducibility, as most studies do not share their code, data or both.

Y. Roy et al. mentioned the importance of EEG analysis. A step in this was handling of artifacts. Jiang, Bian, and Tian did a review addressing literature, which covers handling of artifacts in EEG, outlining methods and applications that address the challenge. They attempted to provide help with determining artifact removal techniques, best suited for necessary requirements given a specific EEG application. Jiang, Bian, and Tian utilized Google Scholar as search engine, with a similar approach as Y. Roy et al., but with different query terms.

The review found over 50% of the studies utilized ICA models or hybrid models, here hybrid models referring combining of classifiers. Their results suggest that no single classifier can succeed at every category, due to high variance across artifact types. These results are agreed upon elsewhere in other fields, like image classifiers in computer vision. A comparative analysis of all the inspected studies reveals different artifact removal methods and their abilities: whether an additional reference exists, whether it works automatically, works online and whether it can perform on a single channel. Table 2 A.1 of the review showcases these abilities in a Boolean form. The best overall methods are based on ICA when used for all types of artifacts, which is also seen in the greatest amount of artifact removal studies utilizing ICA. Given specific circumstances in the setup of the EEG signal, however, can promote other methods for best performance.

Jiang, Bian, and Tian conclude that artifact removal is not a one-size-fits-all solution. Expert knowledge is needed still to produce the best serving methods for specific problems. They point towards further work to combine machine learning algorithms with traditional tools, if such a solution can be found. For now, they provide insight into which methods are applicable, given a set of options.

1.2.1 Current challenges

The reviews by Y. Roy et al. and Jiang, Bian, and Tian both address the EEG community's existing challenges. Often discussed challenges include domain specificity, academic to real-life application, a high learning curve for new members, resource issues (e.g. data availability or expert dependence) or the need for a benchmark to test against, and solutions are constantly proposed. Of the reviewed work, the following selection of solutions are interesting to inspect further for their specific approaches to handling some of the above challenges.

Domain specific

IC_MARK and ICLabel are interesting artifact classifiers that can be used to lower the signal to noise levels. PREP pipeline and the Swartz Center for Computational Neuroscience (SCCN) Makoto Miyakoshi guideline[1] are two recommendations for preprocessing standardization. Artifact handling and preprocessing heuristics are both domain specific challenges. In fact, EEG's artifacts behave quite uniquely throughout the signal and most EEG tools are dependent on the chosen preprocessing pipeline. One of the greatest challenges for moving tools from academic to real-life is the restriction in training data.

[1] https://sccn.ucsd.edu/wiki/Makoto's_preprocessing_pipeline

Continuous updating of datasets such as the Temple University Hospital EEG Corpora is therefore crucial in the efforts made to generalize models. Reviews such as Y. Roy et al. and Jiang, Bian, and Tian greatly improves the learning curve challenge and more modern solutions are also emerging such as ICLabel's crowd computing tutorial[2] that educates while training their algorithm. Ever improving automatic artifact detection algorithms, like IC_MARK and ICLabel, can improve the signal prior to the physician's inspection or be used as full end-to-end approaches. Either way, this reduces the amount of experts required and lowers the barriers of entry, suggesting a general reduced dependency on experts. Finally, benchmarks are emerging, and a simple benchmark implementation for TUH SZ and TUH AR has already been performed by S. Roy et al. in [S. Roy 2019 and S. Roy et al. 2019].

Handling of artifacts during EEG analysis is in of itself a big topic in EEG as Jiang, Bian, and Tian points out. One of the mentioned studies by Frølich, Andersen, and Mørup showcases how to utilize ICA to effectively classify impactful EEG artifacts. They provide 14 effective features for ICA when performing classification, effectively splitting the EEG signal into: brain/neural, eye-blink, Electrooculography (EOG), muscle and heart, reducing the overall signal-to-noise ratio usually found in pure EEG. Their method is still highly dependent on correct preprocessing of the data prior to application and run into trouble if large amounts of the signal is absorbed by a "mixed-signal" classifier. The study still performs adequately with a balanced accuracy of 0.90 and 0.95 for intra-study by two classifiers of artifacts or non-artifacts. The study released the classifer as IC_MARC_SF/IC_MARC_EF. The IC_MARC is proposed to run as an online method running at a 6 min delay. The author suggests using the classifier to detect artifacts observed in the EEG during acquisition, the classifier can then alarm researchers if a critical mass of artifacts is found in the recordings.

IC_MARC were challenged in 2019 by Pion-Tonachini, Kreutz-Delgado, and Makeig, their method ICLabel combined methods from ICA, DL and crowd sourcing for a combinatorial approach, thus performing comparable or better than IC_MARC for several setups presented in both studies. The combination in ICLabel revolves around a semi-supervised GAN network with integrated convolutional layers, the network uses ICA scalp topographies, power spectral densities and autocorrelation functions as input. Crowd sourcing were utilized to assist in providing annotated ICs to the network. The work concluded with two EEG IC classifiers, ICLabel_lite and ICLabel, they performed better or comparable to previous methods, while requiring a substantial lower compute time, as they accomplished classifications at 120ms and 170ms respectively.

The heuristics of EEG preprocessing is often discussed, often resulting in ad hoc processes for the given study or project. These approaches then depend highly on the individual experts executing the preprocessing step of the study or project. SCCN have given a detailed reasoning for preparing EEG data for approaches such as ICA. The preprocessing method of the well-renowned EEG scientist Makoto Miyakoshi is listed on the SCCN website to help emerging EEG researchers understand preprocessing steps when handling EEG signals. Bigdely-Shamlo et al. have proposed a standard preprocessing implementation scaled to large datasets, and recently the pipeline has started development for python's MNE package.

Learning curve

Barriers of entry is a challenge to any academic field. Though, for EEG, challenges due to barriers of entry can be amplified, simply because the field interacts across several

[2] https://labeling.ucsd.edu/tutorial/overview

subjects, ranging from medicine, engineering, computer science or mathematics. It is vital that community members keep in mind that communication should be clear for all the types of EEG researchers. All new EEG researchers are required to have a basic understanding of all subjects that impact the field of EEG. The EEG community has already mobilised to counteract the high barriers of entry. Popular in any field is releasing topic reviews that cover current trends, update recommendations and focus information of importance to the field. Reviews, such as Y. Roy et al., succeeds immensely in meeting the barrier-of-entry challenge by being written with newcomers in mind. EEG tutorials, such as SCCN's tutorial on Makoto's preprocessing guideline, welcomes newcomers by guiding them through the logic and justifications of the more complex steps existing in the field.

More modern approaches are also emerging, and the crowd sourcing used for ICLabel makes it natural to implement educational tutorials, designed to bring its users up to speed. A step included in the work when publishing [Pion-Tonachini, Kreutz-Delgado, and Makeig 2019] [3]. Neuroscience communities like NeuroTechX are also deploying new 'hackatons' for more exploratory learning and bring together the skills and experiences of new and experienced EEG researchers. An expansion of these hackatons are the so-called 'challenges' – competitions that the EEG community have seen an increased amount of in the past years. Just in 2020, Novela Neurotech and NeuroTechX collaborated in introducing the Neureka 2020 Epilepsy Challenge, recommending articles that would be useful when entering the challenge and showcasing the updated TUH SZ corpus. The NeuroTechX community also compiles databases, containing extensive recommendations for newcomers to EEG research.

Data availability

Some of the most common public datasets have been the "EEG Motor Movement/Imagery Dataset" from Schalk et al. at PhysioNet or the extensive European Epilepsy Database has known difficulties. The problem of Schalk et al. stems from the small dataset consisting of just over 1500 one- and two-minute EEG recordings from 109 individual volunteers. This dataset was sufficient for pure EEG analysis with analytical tools. However, when utilizing DL methods, dataset size is of utmost importance for the algorithms to explore the given feature space extensively. The EEG Motor Movement/Imagery Dataset is regrettably unfit for most DL algorithms, due to data quantity. These algorithms are well known to train over thousands or millions of individual samples. Ihle et al. provides high quality data through the European Epilepsy Database. The database is one of the largest, consisting of more than 250 individual patients and 2500 annotated seizures collected in more than 45,000 hours of EEG. Ihle et al. also ensures gold standard annotations from EEG experts that work on clinical EEG data. All these factors make the European Epilepsy Databases one of the gold standards for EEG data. However, the datasets come with a high price tag of 3000 EUR to access 30 patients from the database, meaning researchers get only partial access to the database. [Schalk et al. 2004 Ihle et al. 2012]

Progress happened in 2016 when Obeid and Picone released a large dataset (corpus) from the Neural Engineering Data Consortium (NEDC), established at Temple University. The project has produced the Temple University Hospital EEG Corpus, which became fully available to the public in 2017 on TUH EEG Corpus. The full corpus had over 20,000 EEGs at release, with the goal of expanding the corpus with 1,000 EEGs per year, aiming for 100,000 EEGs in the final corpus. The corpus has been expanded since 2002 by NEDC, and more than 30,000 EEGs are now collected in the corpus, as of this thesis. The corpus has been subset into corpora. The main subset is the TUH EEG Corpus

[3] https://labeling.ucsd.edu/tutorial

(TUEG) containing the full dataset, while further corpora are specialized for unique EEG problems. The TUH Abnormal EEG Corpus (TUAB) includes data annotated as normal or abnormal. The TUH EEG Artifact Corpus (TUAR) has data annotated the five different artifacts of eye movement, chewing, shivering, channel-setup [electrode pop, electrode static and lead artifacts], and muscle. The TUH EEG Epilepsy Corpus (TUEP) offers subjects annotated as epilepsy or without-epilepsy in an even 100 to 100 distribution. The TUH EEG Events Corpus (TUEV) includes EEG segments annotated into the following six different event-classes: (1) spike and sharp wave (SPSW), (2) generalized periodic epileptiform discharges (GPED), (3) periodic lateralized epileptiform discharges (PLED), (4) eye movement (EYEM), (5) artifact (ARTF) and (6) background (BCKG). The TUH EEG Seizure Corpus (TUSZ) displays annotated data for specific seizure types. A detailed annotation guideline was included with TUSZ's release. NEDC published the guideline in 2020 and will publish an official journal in 2021 in The Institute for Signal and Information Processing Report. The TUH EEG Slowing Corpus (TUSL) with annotation for slowing type of events. [Obeid and Picone 2016 Ochal et al. 2020]

Academic to real-life adaptability

It is often mentioned that for deep learning methods to succeed, large datasets with several thousand samples are required. Though, historically, EEG data have been difficult and expensive to collect in such quantity and quality. For well functional end-to-end or general methods to exist, thus, a considerable amount of data acquisition is required. Of the studies Y. Roy et al. inspected, more than half of them contained fewer than 13 subjects, though the authors mention two studies that contain more than 10,000 subjects. One study utilized the dataset acquired by Quan et al. to score sleep stages [Sors et al. 2018]. The dataset has subjects included in numbers rarely seen before. The second study utilizes the already described TUH TUEG corpora, specifically the TUAB corpus, to implement a hidden Markov model, a deep learning context analysis model and a statistical language model hybrid. The hybrid model automatically analyses existence of abnormalities in clinical EEG [Golmohammadi et al. 2019].

The enormous TUH EEG corpora have already impacted the ability of the community to implement models for real-life applications. Though, prior to the release of the TUEG corpora, successful approaches were transitioning from academic to more real-life application. Automatic ICA methods such as IC_MARC have been implemented in the popular MATLAB toolbox, EEGLAB, though the method is constrained by runtime. ICLabel dealt with this constraint to such an extent that the model can now be executed for online EEG setups [Pion-Tonachini, Kreutz-Delgado, and Makeig 2019]. ICLabel are now being used as the default method for automated detection of artifactual ICA components in EEGLAB [Delorme and Makeig 2004].

A completely different challenge regarding academic to real-life application is cost of application. In 2017, McKenzie et al. reported in Nature that more than 80% of the 50 million people dealing with epilepsy were living in low- and middle-income countries (LMICs). Since costly EEG recordings are a standard procedure in the diagnosis of epilepsy in high-income countries, it would be natural to develop a less costly LMIC EEG procedure. Antiepileptic drugs (AEDs) for epilepsy are cheap even for LMICs, making the actual diagnosis the largest barrier for these countries to overcome in treating epilepsy patients. Part of this diagnosis are data recordings from which the physician interprets the patient's brain waves. Even with an availability of 63% and 71% in low-income and lower middle-income countries respectively, only 40% of physicians from low-income countries perceive EEG as affordable [McLane et al. 2015]. Therefore McKenzie et al. proposed to bring an affordable EEG system to the LMICs. The proposed system was a smartphone-based EEG

system with 14 detection electrodes.

Benchmarks

The last two years have expanded the utilization of TUEG. In 2019 the TUAR corpus received a benchmark by S. Roy, who applied a straightforward procedure, thus establishing benchmarks for standard classifiers. As of this thesis, the benchmark is still unchallenged and can be considered state-of-the-art. The benchmark has some issues worth inspecting. One of these is the best performing model, the Linear Discriminant Analysis (LDA), which outperforms a Multi-layer Perceptron (MLP). Y. Roy et al. proclaimed DL methods have shown promising results for EEG analysis, processing and classification and is correlating with the experience Computer Vision had with progression by application of DL models. It seems straightforward to assume that the future will show DL models to outperform a linear model on the non-linear EEG signals. A second issue is the preprocessing pipeline utilized in the benchmark. The benchmark deviates from suggestions such as Makoto and the PREP pipeline for general heuristics when preparing large datasets for EEG analysis. When comparing to the suggestions of Y. Roy et al. regarding reporting of EEG handling or DL reporting, S. Roy deviates by sparsely reporting the preprocessing heuristics. Despite issues in the work of S. Roy, a novel benchmark has been provided to the community enabling comparison work in the future. [Computational Neuroscience n.d. Bigdely-Shamlo et al. 2015]

2020 brought several expansions of TUEG to the community. Among these are the previously mentioned annotation guidelines by Ochal et al., originally produced to the TUSZ corpus, containing general annotation methodology for the entire corpora and the corpora annotation structure. TUSZ was also the focus of the Neureka competition of 2020. Provided by Neurotech and NeuroTechX Neureka asked its participants to predict seizure types from the TUH EEG Seizure corpus. The winners were Biomed Irregulars from KU Leuven, Belgium, who utilized 16 channels to reach the highest Time-Aligned Event Scoring (TAES) of 2.46 used in the competition. The Biomed Irregulars' detection algorithm is a fusion of multiple attention U-nets, whose output is used by an LSTM network for prediction. [Chatzichristos et al. 2020 Neurotech and NeuroTechX n.d.]

2 Theory

2.1 Electroencephalography

A common method for measuring and monitoring brain activity is Electroencephalography (EEG). It is an electrophysiological method to measure the electrical perimeter-activity of the brain. Typically, electrodes are placed on top of the head, in a noninvasive manner. The electrodes will then each observe the electrical stimuli across the cerebral cortex and are connected by one or several reference electrodes. Modern setups install the single electrodes in a cap, to standardize the electrode placement across measurements and patients. The 10-20 system is the internationally recognized setup for electrode placement. The placement of the electrodes is here defined by the distance between adjacent electrodes. The electrodes cover the entire scalp with a 10% distance from front to back and a 20% distance from right to left, totalling in 21 electrodes, as illustrated in figure 2.1.

(a) The original standardized 10-20 electrode setup

(b) The expanded standardized 10-05 electrode setup

Figure 2.1

A modernization of the system has expanded the amount of electrodes by the same thought process to 10-10 and 10-5 systems, increasing the EEG signals' spatial resolution, signal-to-noise ratio and singular electrode dependency etc. Thus, the 10-10 and 10-5 systems have increased the electrode amount to 74 and 345, respectively. Oostenveld and Praamstra 2001

EEG has been a powerful method to diagnose conditions such as epilepsy, sleep disorders, depth of anaesthesia, coma, encephalopathies, and brain death. Jiang, Bian, and Tian 2019 Y. Roy et al. 2019

EEG's price and noninvasive nature compared to other available bio-imaging techniques, such as magnetoencephalography (MOG), magnetic resonance imaging (MRI), and computed tomography (CT) make EEG a favorable solution for diagnostics of the previously mentioned conditions, and EEG is today the preferred method in high-income countries. Research is emerging for more portable EEG systems, lowering entry barriers for countries that do not have access to the method yet. Especially LMICs are interested in a cheap, portable solution. In 2017, McKenzie et al. released a phone-based system, which fills these requirements.

McLane et al. reports in a 60-question survey in 2014 the availability of several biomedical imaging acquisition methods in 37 countries. They recognized a severe need for biomedical imaging in LMICs, as most people suffering from epilepsy are located in LMICs. Of

the 37 surveyed countries, 63% of the low-income and 71% of the lower middle-income countries replied an access to EEG for diagnosis. Furthermore, however, only 40% of physicians from low-income countries perceived EEG as affordable.

WHO reported 2004 of a large disparity of Data collection systems between HICs versus LMICs Organization et al. 2004. These numbers align with the new findings of McKenzie et al. EEG is a commonly available brain activity measurement method in HICs and is a preferred method for diagnosing neurological seizures, aiding physicians to diagnose conditions like epilepsy, narcolepsy, etc. This accessibility enables proactive- or immediate-actions to the diagnosis so people dealing with the conditions can live close to a regular life Organization et al. 2004. Several communities: grassroots, researchers, physicians, WHO, to name a few, spread across the world, are moving towards improving the gab for diagnosis accessibility Sokolov et al. 2020 McKenzie et al. 2017.

2.1.1 EEG for Epilepsy

Epilepsy is a neurological condition, mainly described from the person's tendency to suffer from recurrent epileptic seizures. As per the International League Against Epilepsy [ILAE], epilepsy, in the current understanding, is a direct result of a genetic defect(s) in which seizures are the core symptom of the disorder. A seizure is defined as an abnormal electrical perturbation from the brain; it is often experienced as irregular muscle spasms. [Berg et al. 2010] Individuals are classified as epileptic, if they experience any of the following:

1. Two or more unprovoked or reflex seizures less than 24 hours apart.
2. One unprovoked or reflex seizure combined with a probability of further seizures similar to the general recurrence risk of at least 60% after two unprovoked seizures occurring over the next 10 years.
3. A diagnosis of an epilepsy syndrome.

2.1.2 The mechanisms of EEG

EEG, as a method, monitors electrical signals from the cerebral cortex. Electrical activity is produced by billions of neurons that make up most of the brain. Neurons are electrically charged with dendrites as receivers near the cell body and an axon to propagate the neuro-response to the synapse that transmits to other tissue or dendrites. The brain roughly functions by having neuro synapse patterns. These patterns can then be repeated rhythmically for higher recognizability in the brain. A pattern from a single neuron is too weak to be observed by an EEG electrode. Therefore, EEG activity reflects a summation of the synchronous activity of several neurons' transmission in neuro synaptic patterns.

2.1.3 artifacts in EEG

Artifacts in EEG are elements present in the signal non-neural in nature. They contaminate the EEG recording quality by obstructing the signal of interest, thus increasing the signal to noise ratio. Artifacts are commonly classified as either physiological, some often present being eye movement, muscle activity (e.g. shivering, chewing), or heart beat, or external seen as line noise, electrode signal to noise issues.

Some artifacts can be resolved by notifying data recorders of an issue. Though EEG analysis have started using mathematical tools such as deep learning networks or independent component analysis (ICA), to digitally remove or reduce the artifacts.

3 Methodology

The methodology presents the pipeline used throughout all experiments and results to reproduce the S. Roy benchmark. It describes the usage of TUAR and its descriptive statistics. TUAR is a subset of the public TUH EEG corpora, so an SCCN Makoto guideline fused with PREP pipeline is performed as EEG processing prior to model building. Model building was performed with the hyperopt meta-algorithm. All models were scored against two baselines, the original benchline and an expansion of the original performance metrics.

All code was executed in python3. The popular python EEG library MNE was used as the main EEG-processing tool [1].Gramfort et al. 2014 The PREP pipeline and SciPy was selectively used in addition to MNE for identifying and interpolating bad channels or performing fast-fourier-transform (FFT) to calculate corresponding spectrograms. Scikit-learn, hyperopt and hyperopt-sklearn were used for machine learning purposes to implement models, parameter- and hyperparameter-optimization. Data was handled with numpy, pandas and pytorch, after MNE has performed feature extraction.

3.1 TUAR

Part of the reproduction process of S. Roy uses the same dataset. Their focus was purely on the TUH EEG Artifact Corpus version 1.0.0 (v1.0.0), which was released December 2018. In June 2020, the corpus was updated with annotations and beginning of a version 2.0.0 (v.2.0.0) update, which was released January 2021. v1.0.0 was an early release with only selective event annotation, while v2.0.0 has senior-annotator confirmed annotations for all portions of the signal. This makes v2.0.0 a more complete version with higher descriptive statistics and detailed annotation areas. The benchmark used the v1.0.0 release of the dataset, thus, this work chose the same version.

TUAR consists of 310 recordings (.edf files), which were collected over 270 sessions from 213 individual subjects. The corpus contains 359936 seconds of EEG recordings. TUAR has five artifact annotations: chewing [chew], electrode pop (including electrostatic- and lead artifacts) [elpp], eye movement [eyem], muscle artifacts [musc], shivering [shiv] and two catch labels of background events [bckg] and non-artifacts [null] which are grouped together in this reproduction. All TUEG corpora subsets follow the same labels across corpora, see Appendix A.1. The corpora tightly follow a set of symbols and are developed to follow terms accepted by the EEG research community. Thus, the symbols will only change in case the community agrees upon adding or interchanging terms or symbols. The TUAR corpus sessions are structured in several files:

- .edf: the EEG data in European Data Format.
- .txt: the EEG report corresponding to the patient and session.
- .tse: time-synchronous event file for term-based annotations.
- .lbl: hierarchical graph in a label file for event-based annotations.

The annotations in the corpus can be used in two formats. The .lbl is event-based, meaning annotations of start time, stop time, and seizure type on a specific channel; the .tse is term-based, meaning all channels share the same annotation, corresponding to a start

[1] https://mne.tools/stable/index.html

Label	Total	chew	elpp	eyem	musc	shiv	null/bckg
Subjects	213	22	80	140	75	14	213
Sessions	270	23	97	166	90	14	270
Files	310	28	105	177	99	15	310
Seconds	359,936	2,795	2,647	6,939	5,009	1,245	341,299
Segments	1,466,979	11,884	12,531	32,513	21,671	6,191	1,382,189

Table 3.1: descriptive statistics of EEGs in TUAR v1.0.0

and stop time, which is an aggregation of the above mentioned specific channel annotations. The time-synchronous event [.tse] format is straight forward to read as this example shows: (0.0000, 490.0000, bckg, 1.0000), where the positions indicate the label's: start time in sec, stop time in sec, label and label probability. The annotators use an ACNS TCP montage setup of 21 channel references with 19 unique electrodes when inspecting the EEGs.

Labels across subjects, sessions, recordings and seconds can be seen in Table 3.1. It should be noted the sum of label seconds does not equal the total amount of seconds due to floor-rounding of the label seconds. Of the 310 recordings, the meta data showed the following demographic distribution: 152 were male, 158 were female and 0 were "other". The subjects' age ranged approximately to a normal distribution with mean 52.48 (median 50) between 16 and 95.

TUEG are all recordings of clinical EEG, resulting in a vast variety of channel configurations. The TUAR corpus have reduced this to two types of configurations: average reference (AR and AR_a) or linked ears reference (LE). The montage setup of AR, AR_a and LE can be seen in Figure 3.1. AR and AR_a differs by AR_a not having channels between A1-T7 and A2-T8.

TUAR is one of the most accessible datasets when quantity and quality both are taken into consideration, and together with the reproducibility of the benchmark it is an attractive data set to use.

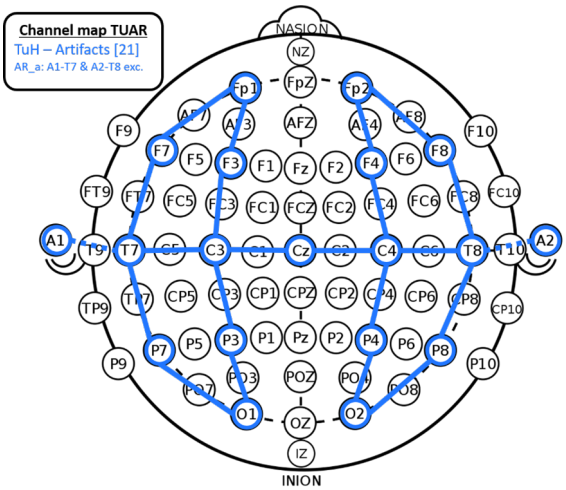


Figure 3.1: Montage setup of TUAR, edges are channels, circles are electrodes: edited from Wikipedia n.d.

3.2 Preprocessing

Collected in a python-dict as a hierarchical structure of subject_subjectID_recording: When loading TUAR, the directory containing the corpus was crawled for every edf-file. A python-dictionary was used to sort the EDF-files into a hierarchical structure, where the keys are three levels of: subject, subject ID and directory to the EDF-file or illustrated as subject_subjectID_recordingPath, this ensures subjects access and enables easy split into training, validation, and test sets further down the methodology. Each recordingPath stores the MNE-loaded raw EDF-file and, at the same, meta data related to original sample frequency, start of recording, end of recording, subject age, subject gender, and all annotations for the recording were stored.

Channel usage in 10-20 system: The TUAR consisted of 30 to 36 channels, of which 19 EEG channels were used to form the TUAR montages, the rest were omitted since they were not used in the annotation scheme of TUAR. The 19 channels Fp1, F7, T3, T5, F3, C3, P3, O1, Cz, Fp2, F4, C4, P4, O2, F8, T4, T6, A1 and A2 were selected to make montages for annotations in TUAR. TUH uses an unconventional naming of the electrodes, <EEG “10-20name”-REF>, a regular expression is used to rename all channels to their conventional 10-05 names.

3.2.1 TUAR EEG processing

TUAR was preprocessed according to Computational Neuroscience guidelines in combination with the PREP pipeline of Bigdely-Shamlo et al. These two methodologies are usually used for preparing and analysing EEG for e.g. ICA or data availability. The suggested pipeline, according to SCCN, follows 15 steps, where the first nine apply to general EEG preprocessing and the last six steps apply to ICA.

All electrodes not included in the 19 selected were excluded prior to the preprocessing sequence.

Step 1, which suggests ensuring data precision of double float, specifically for ICA application, was omitted. Step 2 loads a recording and is handled by MNE for every recording. Step 3 covers downsampling. Following the guideline recordings were downsampled to 250Hz for all recordings to ensure overall common sample frequency. TUAR has variable sample frequency including but not limited to: 250Hz, 256Hz, or 512Hz. Step 4 applies the first filters, the default suggestion is a high-pass filter at 1Hz, though, the filter was replaced with a band-pass filter to focus the signal on the artifact impacted frequencies. The filter-setting has a low-cut frequency and a high-cut frequency of 1Hz and 40Hz, respectively, with a finite window impulse response filter (“firwin”) from the scipy library. Step 5 assigns the recording montage. The electrodes are set up in the MNE ‘standard_1005’ channel setup with a spherical electrode to electrode distance of 95mm. It should be noted that the montage deviates from TUAR and S. Roy, who use a transverse central parietal (TCP) montage resulting in a 22-channel output. The montage deviation happens due to the way in which electrode reference is handled. The TCP-montage pairs selected adjacent electrodes to each other, and such a pairing is called a channel. The deviation found in this thesis pairs every electrode to the global reference as the montage channels. The montage setup results in a 19-channel output, constructed from the same electrodes as the TCP. Step 6 should remove basic line noise. TUAR is recorded in USA, so a “firwin” notch filter of “hamming”-type is applied at 60Hz with a 5Hz filter width. Step 7 suggests removal of “bad”-channels (bads), here defined as: any channels with such a high variance to its reference electrode that rejection of 5-10% of the datapoint does not improve the channel. The PREP pipeline was used for detection and removal of bads by a RANSAC approach. Step 8 interpolates the removed bads with neighbouring channels

by a spherical splines method, which was done by MNE. Step 9 covers re-referencing the recording to the global mean-average for all electrode completing the EEG preprocessing by MNE.

Part of S. Roy EEG preprocessing is vaguely described or omitted, resulting in deviations by interpretations supported mainly by the SCCN guidelines and PREP pipeline. Computational Neuroscience n.d. Bigdely-Shamlo et al. 2015

Assuming S. Roy follows the preprocessing of Schindler et al., it can be assumed the deviations could be narrowed down to: undocumented bandpass filter settings, montage change, notch filter application and downsample frequency.

The bandpass filter S. Roy used is unclear, but it could be interpreted that it was run with a different low- and high-frequency cut at 0.1 and 70 respectively, the SCCN guideline was followed here for the chosen cut frequencies. S. Roy uses the same montage as TUAR, the deviation was chosen according to the SCCN guideline and machine learning approach from TUH utilizes a global reference montage in some implementations, too. It is unclear whether S. Roy uses any line noise removal or not, but following the guideline of SCCN a simple notch-filter was applied well knowing it might remove some line-noise artifacts. S. Roy does not specify any downsampling, though one was applied to standardize all data to an equal sample frequency, reduce recording size and reduce computation time.

3.2.2 TUAR feature extraction

The features extracted for the algorithms input vector was a spectrogram (power density spectrum for 2D), constructed in window segments of the EEG recording. Each segment was constructed as a $1s$ temporal-snapshot of the EEG, segments were translocated in steps of $0.25s$ (75% temporal overlap) along the time axis. The final segment was constructed by $0.25s$ lookback from the last sample, t_N, resulting in varied overlap from the adjacent segment and could be <100% but not 100%.

If the 0.25 seconds did not match the signal's sample frequency, a floor-rounding was performed increasing overlap and ensuring sample integer for segments. The 250 sample frequency was run of $0.248s$ steps (75.2% overlap).

This was performed as data augmentation to extract more input vectors with artifact annotations, the resulting increase from data augmentation can be seen in the last row of Table 3.1. S. Roy reports a 75% overlap when reporting FFT, but it was unclear if this overlap was temporal, frequency or both in nature, though it was interpreted as both.

The spectrogram of each segment was calculated using the scipy spectrogram function, using a 'tukey' window with alpha 0.25 in 'psd' mode and 75% FFT overlap. The spectrograms were z-score normalized and cropped at 45hz across all channels. All spectrogram would have a corresponding label, recovered by the '.tse' TUAR file. The annotations existing in the segment were assigned as label in the segment, resulting in some segments having several labels. The spectrogram and labels were saved to the hard drive as a pytorch tensor with the spectrogram as float16 and label as a string-array to lower storage size.

Each tensor could be loaded individually as needed into a (X,Y) setup commonly used for scikit-learn. A spectrogram were of the size $[channel \times frequency-axis \times time-axis] = [19 \times 46 \times 1]$ and was reshaped to a vector of $[channel \cdot frequency-axis, time-axis] = [874, 1]$. A label was loaded by its string-value to a $[6, 1]$ sparse hot encoding according to ['eyem', 'chew', 'shiv', 'elpp', 'musc', 'null']. Thus, a single example of (observation

All ar subjects	train split	val split	test split	Under-sampled	train mini	val mini	test mini
eyem	14423	4681	13329	>	4418	1316	3659
chew	5386	2780	3473	>	850	446	393
shiv	5236	176	552	>	2089	44	221
elpp	6685	1711	3473	>	1626	792	1340
musc	10204	3907	4289	>	1555	577	710
null	31036	10010	19472	>	10268	3009	6040

Table 3.2: Undersampling of the TUAR to enable experiments

features X, target Y), would be 874 features with an [0, 1, 0, 0, 0, 1] array signifying a chewing and null annotation.

3.3 Modelling

After all subjects were loaded with its window segments, the rest of the method was run with a set randomized seed for reproducibility. Thus, whenever “random” is used in this thesis, it was done with set seed. The subjects were randomly segregated into train-, validation- and test-set of (50% / 15% / 35%) data splits. All feature vectors containing NAN-values was removed from the data sets. Due to gross overrepresentation of null annotations, an undersampling strategy was performed. Observations only containing null was removed by a factor of 30 ensuring observations containing null and artifacts were not undersampled. This null undersampled data set was referenced as the full dataset. The null undersampling factor was not documented by S. Roy so an alternate source was found. Kim and Keene used 30 as factor, and this factor contains null as a majority label while empowering the artifact labels as a trade-off.

Due to limited computation power the dataset was constrained to only include the average reference session ‘ar’, the ‘ar_a’ and linked ear session was excluded. Furthermore, the ‘ar’ subjects were undersampled with weighted priors on all labels; labels with high representation were undersampled heavier than those with low representation. Table 3.2 show the label representation prior and post undersampling.

It should be noted the data has a total more labels than S. Roy 2019, which will be discussed later. The undersampled data was randomly shuffled before being fed to the machine learning algorithms.

3.3.1 Model building with hyperopt

Models were parameter tuned by hyperopt in a scikit-learn inspired framework hp-sklearn linked with the hyperparameter tuning algorithm hyperopt.

Hyperopt is a meta-algorithm, implemented as a Python library, that perform serial and parallel optimization for popular algorithms to explore the more challenging search spaces. S. Roy produced their benchmark for eight algorithms. When reproducing the benchmark for this thesis, only seven hyperopt algorithms were supported and therefore included in the selection. The seven algorithms were: adaboost, gauss naïve bayes (GaussNB), k-nearest neighbour (knn), linear discriminant analysis (LDA), random forest, sklearn’s stochastic gradient descent (SGD) and XGBoost. The Multi-layer Perceptron (MLP) was excluded due to no hyperopt-sklearn implementation and S. Roy does not specify the setting of the perceptron. [Bergstra, Yamins, and Cox 2013]

Hyperopt was initialized with the given classifiers. The loss function used a minimized f1-score (i.e 1-f1_score), and six cores were utilized in hyperopt by setting n_jobs = 6. The concatenated training and validation set, X_model and Y_model, was fed to the classifiers. Hyperopt used the validation set to explore and find the optimal hyperparameters for each classifier, training was done without cross-validation shuffle and set_seed.

All classifiers were trained in a one-vs-rest (one-against-all) setup resulting in five train/-val/test setups for each classifier. A window can cover several labels, which are included for the data fed the classifiers by the construction of Y when presented to the classifiers.

The hyperopt model estimator utilizes the sklearn base estimator as a constructor for the object. The class object has several definitions. When initialised we use the estimator to optimise the search space of seven classification algorithms. Hyperopt is initialised by defining model inputs in three types: fixed-default settings, variable-default settings, user settings.

Fixed-default settings are not changed when initializing the object. variable-default settings are settings in which hyperopt searches the search space of defined algorithm. The rest of the settings that we define outside the object are called user settings. Fixed-default settings for hyperopt:

- .algo search algorithm is run as random search
- .seed random state seed is run with a numpy random seed.
- .fit_increment fixed trials are how trials will be a synchronization barrier for ongoing trials and is one.
- .max_evaluation of each model is set to 10 (5 in roy2019machineTUAR).
- .refit retrains the best models with the full variable X_model data.
- .space as undefined uses the fixed-default settings:
  - .preprocessing is none due to being done prior to hyperopt.
  - .ex_preprocs is none because it is not needed.
  - .n_ex_pps is the length of a list of ex_preprocs which is 1.

Variable-default settings for hyperopt:

- .space searches the chosen classifier's unique search spaces. Each algorithm's search spaces are covered separately later.

User-settings for hyperopt:

- .loss_fn defines the loss function the classifiers are trained for, an adapted sklearn implemented f1-score was adapted by subtracting the score from 1.
- .n_jobs allows hyperopt to utilize several CPU kernels, this is set to 6
- .classifier defines the machine learning algorithm hyper parameters that will be explored, we use the hyperopt one_vs_rest constructor for each run.

The sklearn one_vs_rest constructor is usable in hyperopt where the seven algorithms are defined as a list for running the default hyperopt settings of the algorithm, except for knn and LDA, which additionally are defined with enabled input data as sparse for knn and fixed n_components as 1 for LDA to enable one_vs_rest runs.

All algorithms use the equivalent sklearn algorithm constructor, except for XGBoost, which has its own constructor. The unique search spaces of the algorithms are then explored by hyperopt, which will be described separately.

3.3.2 Adaboost

Hyperparameter exploration in hyperopt utilizes a default search space, which are: a base estimator, number of estimators, the learning rate of the algorithm, a random state seed and boosting algorithm method.

Adaboost is run anew without a base estimator, where a q-log-uniform(ln(10.5), ln(1000.5), step increments of 1) distribution of estimator are explored, with a learning rate of log-normal(ln(0.01), ln(10.0)) for an initial integer random state produced by numpy.random.-RandomState(5) and random select between 'SAMME' and 'SAMME.R' boosting algorithm.

3.3.3 GaussNB

Hyperopt uses a deterministic setup of gaussian naïve bayes in a non-parametric setup. The GaussNB parameters are calculated as: epsilon is the additive value of variances, sigma is the variance of each feature per class, and theta is the mean of each feature per class.

3.3.4 Knn

Hyperparameters are vast, though in hyperopt there is a focus on: neighbours' inclusion, weight neighbour proximity and algorithm method. Hypertop includes q-log-uniform(ln(.5), ln(50.5), step increments of 1) neighbours, weights was given uniformly or distance by their inverse distance and random selecting between four algorithm methods of 'auto', 'ball_tree', 'kd_tree' or 'brute'.

3.3.5 LDA

Two type hyperparameters are explored for the one_vs_rest setup in hyperopt and are the solvers to use and shrinkage approach. Five combinations of solver with shrinkage are randomly selected between: a singular value decomposition (SVD) without shrinkage, leash square with shrinkage or without, eigenvalue decomposition with shrinkage or without. To ensure LDA enables for one_vs_rest, n_components are fixed to one.

3.3.6 Random forest

Hyperparameters that are explored by hyperopt are restricted to: number of trees $n\_estimators$, number of features to consider when looking for best split $max\_features$, the maximum depth of the tree max_depth, the minimum number of samples required to be at a leaf node $min\_samples\_leaf$, to run with or without bootstrapping $bootstrap$ and measuring criterion for quality of a tree split $criterion$.

The following hyperparameters were included as fixed setting, where the minimum number of samples required to split an internal node $min\_samples\_split$.

Default hyperopt optimizes the following hyperparameters:

- $n\_estimators$ is a q-log-uniform(ln(9.5), ln(3000.5), step increments of 1) distribution.
- $max\_features$ is picked randomly by a distribution of [(0.2, 'sqrt'), 0.1, 'log2'), (0.1, None), (0.6, uniform(0, 1)].
- max_depth is picked randomly by a distribution of [(0.7, None), (0.1, 2), (0.1, 3), (0.1, 4)] where (probability, max_depth).

- $min\_samples\_leaf$ is picked randomly by as an integer of one or a q-log-uniform(ln(1.5), ln(50.5), step increments of 1) distribution.
- $bootstrap$ is picked randomly between included or excluded.
- $criterion$ is picked randomly between 'gini' and 'entropi'.

3.3.7 SDG

Hyperparameters that are explored by hyperopt are restricted to: Which type of loss is used $loss$, regularization term $penalty$, constant regularization parameter $alpha$, Elastic Net mixing parameter 1-ratio, stopping criterion tol when loss > best_loss - tol, maximum number of passes over training data max_iter, learning rate schedules between 'constant', 'optimal', 'invscaling' and 'adaptive' $learning\_rate$, initial learning rate $eta0$, exponent for inverse scaling learning rate $power\_t$ and how to handle class balance $class\_weight$.

Default hyperopt optimizes the following hyperparameters:

- $loss$ is picked randomly by a distribution of [(0.25, 'hinge'), (0.25, 'log'), (0.25, 'modified_huber'), (0.05, 'squared_hinge'), (0.05, 'perceptron'), (0.05, 'squared_loss'), (0.05, 'huber'), (0.03, 'epsilon_insensitive'), (0.02, 'squared_epsilon_insensitive')].
- $penalty$ is picked randomly by a distribution of [(0.40, 'l2'), (0.35, 'l1'), (0.25, 'elasticnet')].
- $alpha$ is a log-uniform(ln(1e-6), ln(1e-1)) distribution.
- 1-ratio is a uniform(0, 1) distribution.
- tol is a log-uniform(ln(1e-5), ln(1e-2)) distribution.
- max_iter is a q-log-uniform(ln(1e7), ln(1e9), step increments of 1) distribution.
- $learning\_rate$ is picked randomly by a distribution of [(0.50, 'optimal'), 0.25, 'invscaling'), 0.25, 'constant')].
- $eta0$ is a log-uniform(ln(1e-5), ln(1e-1)) distribution.
- $power\_t$ is a uniform(0, 1) distribution.
- $class\_weight$ is randomly pick between (None, 'balanced').

3.3.8 XGBoost

Hyperparameters that are explored by hyperopt is restricted to: The maximum depth of a tree max_depth, the step size shrinkage for updating when preventing overfitting, commonly called the learning rate $learning\_rate$, amount of rounds where weights are boosted for the ensembled estimators $n\_estimators$, minimum loss reduction required to make a further partition on a leaf node of the tree $gamma$, minimum sum of weights of all observation required in a child min_child_weight, maximum delta step we allow each leaf output to be max_delta_step, subsample ratio of the training instances $sumsample$, the subsample ratio of columns when construction each tree $colsample\_bytree$, the subsample ratio of columns for each level $colsample\_bylevel$, $l1$ regularization term on weights reg_alpha and $l2$ regularization term on weights reg_lambda.

The following hyperparameters was included as fixed setting: the objective was set to a binary evaluation, balance control of positive and negative weights $scale\_pos\_weight$ and the initial prediction score for all instances $base\_score$. The user fixed hyperparameters

Baseline majority	accuracy	acc_{eyem}	acc_{chew}	acc_{shiv}	acc_{elpp}	acc_{musc}	acc_{null}	
	0.833	0.704	0.968	0.982	0.892	0.943	0.511	height

Table 3.3: Calculated majority voting baseline

were: amount of kernels used n_jobs, evaluation metric $eval\_metric$ and random seed $seed$.

Default hyperopt optimizes the following hyperparameters:

- max_depth is a uniform(1, 11) distribution.
- $learning\_rate$ is a [log-uniform(ln(1e-4), ln(0.5)) - 1e-4] distribution.
- $n\_estimators$ is a q-uniform(100, 6000, step increments of 200) distribution.
- γ is a [log-uniform(ln(1e-4), ln(5)) - 1e-4] distribution.
- min_child_weight is an integer log-uniform(ln(1), ln(100)) distribution.
- $subsample$ is a uniform(0.5, 1) distribution.
- $colsample\_bytree$ is a uniform(0.5, 1) distribution.
- $colsample\_bylevel$ is a uniform(0.5, 1) distribution.
- reg_alpha is a [log-uniform(ln(1e-4), ln(1)) - 1e-4] distribution.
- reg_lambda is a log-uniform(ln(1), ln(4)) distribution.

3.4 Model scoring

The models were trained and validated by a weighted f1-score, thus the primarily performance metric will be identical to the benchmark. Accuracy and label weighted-sensitivity was included true to the benchmark, as an expansion of the benchmark was the following metrics included: total balanced accuracy, accuracy per label, and total weighted-sensitivity. Two naive baselines to outperform was calculated, a majority-vote baseline on the test set for overall label accuracy and a permutation baseline (labels are randomly shuffled) on the test set, which was averaged over 1000 repetitions.

The majority-vote baseline was calculated by resulted in table 3.3

4 Experiment

The experiment chapter presents a structure, meant to test, and inspect the methodology. The methodology could broadly speaking be split into three sections: accessing and signal processing TUAR, reproducing the classifiers of the benchmark and evaluation of said classifiers.

The utilized dataset for all experiments was TUAR, it could be accessed by this thesis code at the thesis' repository on [GITHUB]. The signal processing of TUAR could found at the same repository, which presents the overall methodology. DTU-compute have started several student projects with interest in the TUAR corpus, where the student-groups have utilized this repository to initialize their projects. The repository has been accessed by students throughout the semesters of spring 2020 to spring 2021. The repository has been general enough that students have been able to utilize TUAR directly or edit the code to their specific projects. The code run for the thesis can be found in the *thesis* folder in the repository.

Three experiments were run to highlight the reproduction capability of the methodology. S. Roy has a focus on pure model performance. Therefore, the experiments focus on the same area. An argument could be made for focusing on EEG utilization in TUAR, by optimizing on aspects such as: EEG processing, feature engineering, data augmentation etc. which all could impact the benchmark. The sparse explanation of these aspects in the benchmarks limits the capabilities for the thesis to give a true assessment for improvements or reproduction. It should be noted the same limits also could impact the overall model performance of the benchmark reproduction, presented in this work.

Each experiment was an inspection of the methodology and was structured by its approach, reasoning, application, achievement and results. All to be used for the discussion.

All experiments had lower dataset than what was seen in the benchmark due to restricted computation power and time.

4.1 Experiment 1

The first experiment was a first attempt of the methodology and the first attempt at showcasing the reproducibility of the benchmark. General usability of the code implementation from the methodology was inspected too. Cases of issues with specific setups or patients could also be inspected here, which carried over to further experiments.

The trial run implies the methodology was executed as described, producing seven models from the subsampled dataset for all labels. The experiment resulted in Table 4.1, Table 4.2 and Table 4.3, The performance runs of S. Roy can be seen in Table 4.4 Both tables have the best performing model highlighted in bold for each metric.

4.2 Experiment 2

The second experiment was an ablation study of the label impact on the classifiers, the ablation study randomly undersampled the eye movement label of the training set. Class specific learning curves was used to illustrate the impact of under representation of a label. The ablation study was performed to showcase the methodology robustness to underrepresented labels or if a cutoff could be found for how much representation would

Algorithm	wF1	acc	balanced acc	S_{all}
AdaBoost	0.687	0.653	0.274	0.674
GaussianNB	**0.731**	0.577	**0.477**	**0.748**
k-NN	0.687	0.668	0.288	0.664
LDA	0.701	0.660	0.327	0.683
Random Forest	0.694	0.629	0.284	0.687
SGD Classifier	0.687	0.664	0.282	0.669
XGBoost	0.702	0.617	0.297	0.693
Baseline$_{permutation}$	0.663	0.361	0.322	0.663
Baseline$_{majority}$	-	**0.833**	-	-

Table 4.1: overall performance metrics

Algorithm	S_{all}	S_{eyem}	S_{chew}	S_{shiv}	S_{elpp}	S_{musc}	S_{null}
AdaBoost	0.674	0.616	0	0	0.026	0.001	**1.000**
GaussianNB	**0.748**	**0.704**	**0.891**	0	0.143	0.127	**1.000**
k-NN	0.664	0.553	0.000	0	0.041	0.137	**1.000**
LDA	0.683	0.596	0.137	0	0.054	**0.182**	0.994
Random Forest	0.687	0.656	0.008	0	0.028	0.011	**1.000**
SGD Classifier	0.669	0.583	0	0	0.031	0.076	**1.000**
XGBoost	0.693	0.668	0.053	0	0.025	0.035	**1.000**
Baseline$_{permutation}$	0.663	0.573	0.061	**0.035**	**0.210**	0.111	0.945
Baseline$_{majority}$	-	-	-	-	-	-	-

Table 4.2: Sensitivity score of labels

Algorithm	acc_{eyem}	acc_{chew}	acc_{shiv}	acc_{elpp}	acc_{musc}	acc_{null}
AdaBoost	0.695	0.938	0.893	0.790	0.889	**0.945**
GaussianNB	0.688	0.765	0.888	0.763	0.883	**0.945**
k-NN	0.655	0.938	0.877	0.794	0.890	**0.945**
LDA	0.667	0.925	0.890	0.764	0.868	0.940
Random Forest	0.702	0.936	0.894	0.793	0.888	**0.945**
SGD Classifier	0.672	0.938	0.891	0.789	0.891	**0.945**
XGBoost	**0.711**	0.938	0.893	0.791	0.890	**0.945**
Baseline$_{permutation}$	0.510	0.885	0.933	0.669	0.802	0.896
Baseline$_{majority}$	0.704	**0.968**	**0.982**	**0.892**	**0.943**	0.511

Table 4.3: Accuracy scores of labels

Algorithm	Weighted-F1	Accuracy	S_{eyem}	S_{chew}	S_{shiv}	S_{elpp}	S_{musc}	S_{null}
AdaBoost	0.7375	62.57%	62.51%	68.63%	2.31%	28.30%	62.88%	63.17%
GaussianNB	0.7773	67.79%	63.19%	72.67%	**16.32%**	13.99%	43.47%	69.03%
k-NN	0.7476	63.76%	60.77%	**86.07%**	5.95%	26.06%	48.55%	64.67%
LDA	**0.8012**	**71.43%**	58.73%	62.73%	2.50%	26.99%	**70.76%**	**72.39%**
Random Forests	0.7834	68.80%	**73.35%**	80.35%	3.00%	35.26%	67.25%	69.39%
SGD classifier	0.7887	69.57%	63.06%	73.61%	3.10%	28.79%	69.01%	70.36%
XGBoost	0.7996	71.19%	72.38%	74.08%	2.75%	**38.75%**	67.91%	71.90%

Table 4.4: table taken from S. Roy 2019

be needed to classify said label. Only the largest artifact label was selected for ablation, which was eye movement, due to time constraints and computation power available. It was undersampled to a percentage of the original label amount of 4418 to 1%, 5%, 10%, 25%, 50% and 100% for each classifier, [44, 220, 441, 1104, 2209, 4418] included eye movement labels respectively.

The learning curve of each classifier for the weighted F1-score, overall sensitivity and eye movement sensitivity was illustrated in Figure 4.1, Figure 4.2 and Figure 4.3 respectively.

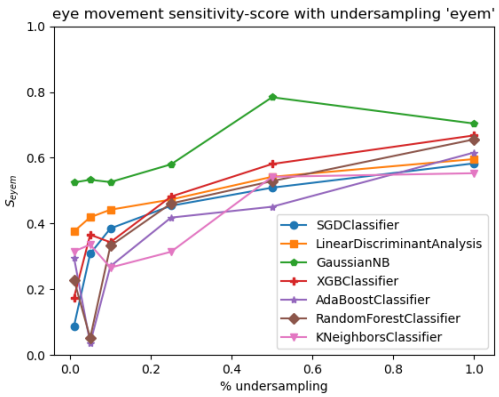
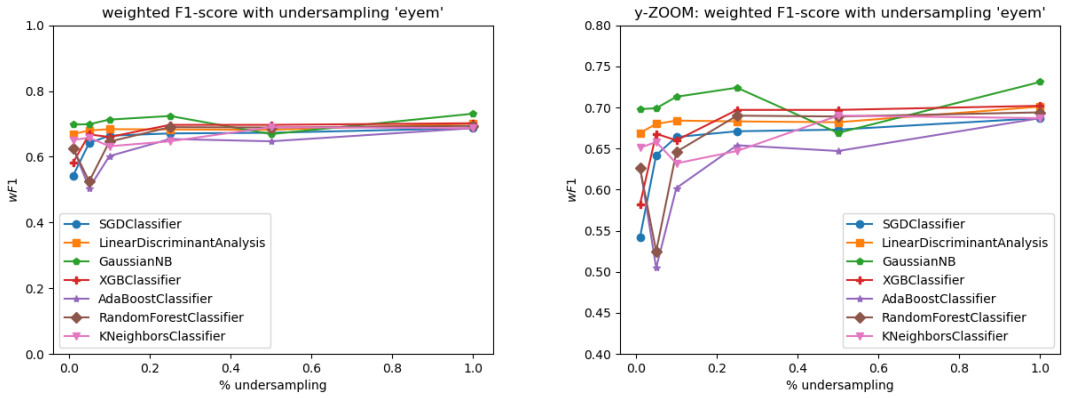


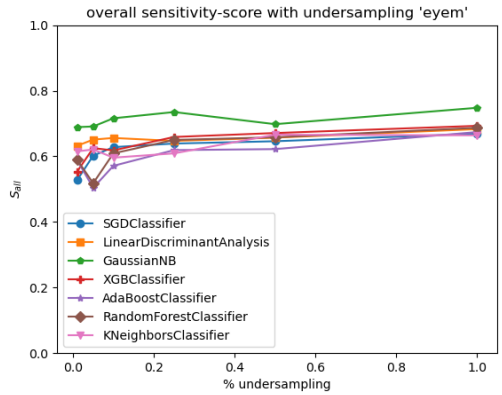
Figure 4.1: Model S-eyem score performance when undersampling eyem



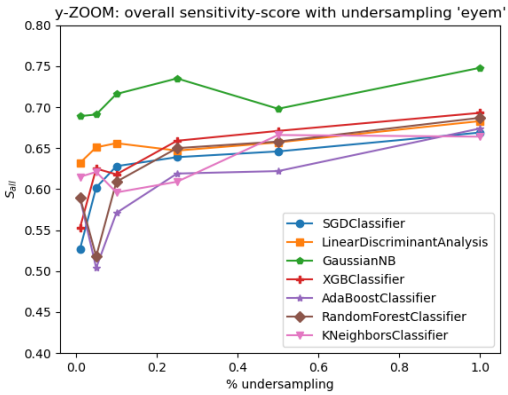
(a) Original figure for wF1-score

(b) Zoomed on Y-axis

Figure 4.2: Model wF1 score performance when undersampling eyem



(a) Original figure for S_{all}



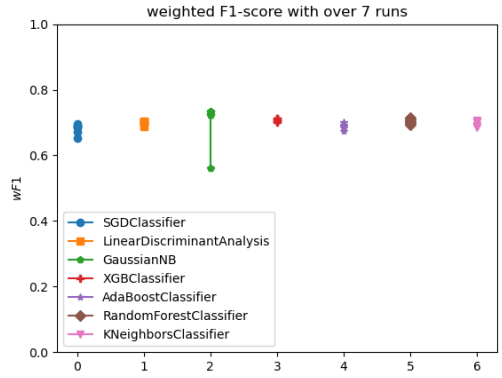
(b) Zoomed on Y-axis

Figure 4.3: Model S_{all} score performance when undersampling eyem

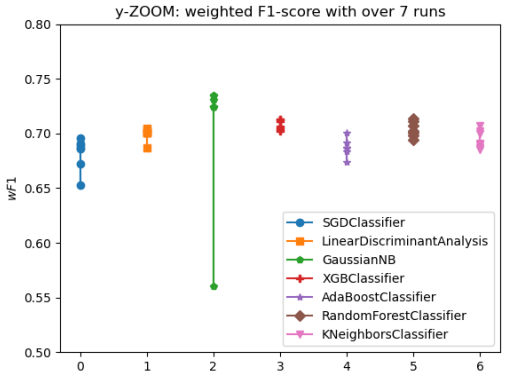
4.3 Experiment 3

The third experiment was repetition of the first one without randomness control. It was performed to inspect how steady the methodology performs, with respect to the inherent randomness of the hyperparameter search in hyperopt for each model. The experiment removes the methods fixation on the set.seed parameter and reintroduces the regular randomness generally existing. The set.seed parameter was a hash seed of the time the code was started and was independently executed seven times for each model.

The experiment produced the illustrations seen in Figure 4.4 for weighted-F1 score.



(a) Original figure for uncontrolled set.seed



(b) Zoomed on Y-axis

Figure 4.4: Model scores for uncontrolled set.seed

5 Discussion

The discussion will focus on the ability of the methodology to produce a pipeline, which could reproduce results shown in S. Roy 2019. To even start an attempt at reproduction, the methodology would need to produce similar window-segments from TUAR processing as seen in the benchmark.

5.1 TUAR processing

Experiment 1 showed that the methodology in general can produce models using the pipeline chosen for TUAR. 310 edf-files was signal processed to produce 1.447.344 (tensor, label) pairings, which could be used to construct a subset of training-, validation-, and testing-data, which approximately follows the original data. The benchline had a total of 1.432.937 segments, when S. Roy reports amount of EEG sessions, the number was lower than the 270 TUAR could provide at the time of downloading the corpus.

The Benchmark does not give a clear reasoning for how the TUAR was processed, enforcing searching for alternative references. The changes to the processing of TUAR falls to the Computational Neuroscience guidelines in combination with Bigdely-Shamlo et al. pipeline, which recommends the chosen approach used in the methodology.

It should be noted that when performing train/val/test split that just 14 of the subjects contain shivering artifacts, which impacted the chance of having shivering artifacts randomly split into correct data splits.

5.2 Experiment 1

When inspecting the Table 4.1 from experiment 1, GaussianNB was the overall best performing model with a weighted-F1 score of 0.731, balanced accuracy of 0.477 and S_{all} of 0.748. When comparing to the numbers reported in the benchmark, seen in Table 4.4, every algorithm performs worse, which could be explained by either the amount of data the models are trained on or that the eigenvalue feature representation performs better. Notably the benchmark achieve the best performance by the Linear Discriminant Analysis algorithm, in contrast to GaussianNB of experiment 1.

A clear issue was noticed for all models ability to detect shivering artifacts, Table 4.2 shows the models could not select shivering artifacts at all The extremely low validation set for the shivering artifact could explain why the models was badly tuned to catch this artifact, as it would be difficult to find correct hyperparameters, which would be sensitive to the artifact.

Several of the models were able to pick out all null artifacts. Indicating the models are tuned too highly to guess on the null label, providing a problem for the usage of artifact detection.

A similar issue can be noticed for the chew artifact where only the GaussianNB and LDA outperforms the permutation baseline. The GaussianNB highly outperforms the baseline and when compared to the benchline even performs better than the 86.07% chew sensitivity of the k-NN.

The best accuracy was in general to guess on the non-label class, indicated by the majority vote baseline performing best for both the overall accuracy and several of the label specific

Algorithm	wF1	S_{all}	S_{eyem}	S_{chew}	S_{shiv}	S_{elpp}	S_{musc}	S_{null}
GaussianNB	0.560	0.561	0.783	0.891	0	0.430	0.762	0.431

Table 5.1

accuracies, when inspecting Table 4.3. It was also indicated here that most of the models prioritizes the null class too much.

5.3 Experiment 2

Figure 4.1 clearly illustrates the impact of low label inclusion, the sensitivity score dwindles for all models the lower the label inclusion was. When inspecting the Figure, XGBoost was one of the classifiers gaining the most by the higher inclusion. The still increasing gradient of by the end of the Figure could indicate that the models still could be improved by giving it more eye movement data.

Though not as extreme, it can be observed the exclusion of a label similarly impacts the overall weighted F1-score in Figure 4.2. The models starts to plateau at 25% label inclusion, which was 1104 label observations. A cutoff amount where the eye movement label obtains a similar range of other labels. It can also be observed LDA was the least impacted model on the weighted F1 score regarding label undersampling. Though, GuasisanNB did not see a large impact either.

Similar observations could be seen, when inspecting the impact on overall sensitivity in Figure 4.3. The figure in general shows the increasing performance with more label inclusion.

The models random forest and AdaBoost, both forest models, performed in substantially lower on their 5% inclusion run in comparision to the 1% inclusion.

5.4 Experiment 3

When inspecting Figure 4.4, the models perform within the same area, GaussianNB has a single run performing under 0.6, while the rest consistently performs better than the rest.

A resason for the lower score from the single GaussianNB run could be explained by its classification of null labels, seen in Table 5.1.

The classifier was the only example seen prioritize the null label so low, which naturally will provide a general lower score as null is the majority label, but its the classifier with the best seen sensitivity scores across the board.

XGBoost performed consistently throughout the runs while the SGD classifier had the highest variety between .65 and .7.

5.5 Benchmark reproducibility

The benchmark proved difficult to reproduce due to several factors, but the highest factor was the absense of a repository to follow the benchmark decisions, leaving the reader to interpret concrete and vague reasoning. The benchmark could also increase reproducibility by applying the points from Y. Roy et al. in their appendix B *Checklist of items to include in a DL-EEG study*. They recommend authors includes decisions regarding, data, EEG processing, model specifics (network architecture), hyperparameter training and performance metrics.

The benchline does a good job of indicating the data usage. Though, crucial sentences are left to the reader, it is never indicated if there is a segmentation overlapping of the one second windows. A decision was made to interpret the 75% overlap as both temporal and spectral as the reported seconds in the TUAR table did not add up else wise. The benchline undersamples the null-label without stating a factor of this, another reference was needed to establish a decision.

General EEG processing was mostly left to the reader for interpretation, most of the EEG processing followed a different guideline due to the opaqueness of the benchline.

All models in the benchmark was included and described though hyperopt, though it was unclear how the hyperopt library was utilized as it does not include specific models, why the hyperopt-sklearn sub-library was utilized. The library does not have default support for MLP models, why it was excluded. There never was a clarification if hyperopt was tuned to the specific problem, why the default setting was assumed.

Performance metrics was clearly reported by S. Roy

5.6 TUAR accessibility

TUAR was an impressive collection of EEG recordings. Its documentation facilitated systematic processing weather it being the raw EEG signal or corresponding annotations.

Through the work of this thesis have the dataset been enabled to DTU-students, which can start an understanding of EEG and ML.

5.7 Future works

The work could reach improved results by applying further experiments to get closer to the original benchline.

First of all, running the code on the full TUAR corpus, would be interesting as the Figures from the experiments indicates the models can be improved by such an inclusion. A ablation study on the other labels could also indicate how much could be gained by data inclusion.

TUAR was processed differently to the benchline. But a study to explore which aspect of the signal processing could make the transition clearer

The hyperopt algorithm states it has settings that could better fit the TUAR problem than the default setting. Exploring the library-settings could prove a more interesting hyperparameter search space for the specific problem TUAR indicates.

6 Conclusion

This thesis attempted to reproduce the benchmark presented in S. Roy 2019 and its application of the recently release EEG artifact dataset, TUAR.

An approximation for processing the TUAR corpus was accomplished, but the details described in the benchmark does not enable reproducing a processing pipeline to a full feature vector, thus an alternative approach from the renowned Computational Neuroscience n.d. and Bigdely-Shamlo et al. 2015, was used for the finished feature vector, still vastly different to the one presented in the benchmark. The feature vector was still based on a FFT-base across the same channels 19 for the TCP montage system. A similar montage system was found with a global mean reference montage, which replaced the TCP montage, the models was constructed with similar FFT features to the benchline in a larger feature matrix that did not depend on correlation coefficients. The thesis made the step of applying the TUAR corpus more transparent.

The models produced by this work, did not accomplish comparable results to the benchmark, due to the opaqueness of the benchmark, it is unclear specifically where difference lie. It was suggested a big contributor to the difference lie in the size of the data set the models are presented, however, the tuning of the hyperopt-meta algorithm could explain part of the difference too. Lastly the previous mentioned application of signal processing on TUAR could contribute too.

The benchmark proved difficult to reproduce due to no repository, forcing scientist to reverse engineer the process, an attempt was made of this process. The attempt could produce results which could be compared to the original benchmark, though, the reported benchmark outperformed all models presented in this work, the models had a over prioritization of the non-artifact majority label 'null', as the main indicator of the lower scoring. Contrary to best model of the benchmark, Linear Discriminant Analysis, did this work produce the best artifact classifier with GaussianNB over several runs.

Including a larger dataset, indicates better performance on eye movement artifacts, as promising results could be obtained by utilizing the full TUAR dataset when trying to reproduce the benchmark using this code, that can be found at [GITHUB].

Reproduction of the benchmark still eludes this work, though a more open attempt was presented. By provided a more expansive methodology for how to handle TUAR and the libraries utilized in the benchmark, could help future interested parties in challenging the benchmark. The code have already been used at DTU to introduce students to the TUAR dataset, and is general enough that students have applied it to other EEG-databases.

Bibliography

Roy, Subhrajit (2019). “Machine Learning for removing EEG artifacts: Setting the benchmark”. In: *arXiv preprint arXiv:1903.07825*.

Plesser, Hans E (2018). “Reproducibility vs. replicability: a brief history of a confused terminology”. In: *Frontiers in neuroinformatics* 11, p. 76.

Roy, Yannick et al. (2019). “Deep learning-based electroencephalography analysis: a systematic review”. In: *Journal of neural engineering* 16.5, p. 051001.

Jiang, Xiao, Gui-Bin Bian, and Zean Tian (2019). “Removal of artifacts from EEG signals: a review”. In: *Sensors* 19.5, p. 987.

Goldberg, Eli, Norbert Driedger, and Richard I Kittredge (1994). “Using natural-language processing to produce weather forecasts”. In: *IEEE Expert* 9.2, pp. 45–53.

Chalapathy, Raghavendra and Sanjay Chawla (2019). “Deep learning for anomaly detection: A survey”. In: *arXiv preprint arXiv:1901.03407*.

Roy, Subhrajit et al. (2019). “Machine learning for seizure type classification: setting the benchmark”. In: *arXiv preprint arXiv:1902.01012*.

Frølich, Laura, Tobias S Andersen, and Morten Mørup (2015). “Classification of independent components of EEG into multiple artifact classes”. In: *Psychophysiology* 52.1, pp. 32–45.

Pion-Tonachini, Luca, Ken Kreutz-Delgado, and Scott Makeig (2019). “ICLabel: An automated electroencephalographic independent component classifier, dataset, and website”. In: *NeuroImage* 198, pp. 181–197.

Bigdely-Shamlo, Nima et al. (2015). “The PREP pipeline: standardized preprocessing for large-scale EEG analysis”. In: *Frontiers in neuroinformatics* 9, p. 16.

Schalk, Gerwin et al. (2004). “BCI2000: a general-purpose brain-computer interface (BCI) system”. In: *IEEE Transactions on biomedical engineering* 51.6, pp. 1034–1043.

Ihle, Matthias et al. (2012). “EPILEPSIAE–A European epilepsy database”. In: *Computer methods and programs in biomedicine* 106.3, pp. 127–138.

Obeid, Iyad and Joseph Picone (2016). “The temple university hospital EEG data corpus”. In: *Frontiers in neuroscience* 10, p. 196.

Ochal, Domenic et al. (2020). “The Temple University Hospital EEG Corpus: Annotation Guidelines”. In: *Institute for Signal and Information Processing Report* 1.1. Self release at https://www.isip.piconepress.com/projects/tuh_eeg/ Journal release 01/04/2021.

Quan, Stuart F et al. (1997). “The sleep heart health study: design, rationale, and methods”. In: *Sleep* 20.12, pp. 1077–1085.

Sors, Arnaud et al. (2018). “A convolutional neural network for sleep stage scoring from raw single-channel EEG”. In: *Biomedical Signal Processing and Control* 42, pp. 107–114.

Golmohammadi, Meysam et al. (2019). “Automatic analysis of EEGs using big data and hybrid deep learning architectures”. In: *Frontiers in human neuroscience* 13, p. 76.

Delorme, Arnaud and Scott Makeig (2004). “EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis”. In: *Journal of neuroscience methods* 134.1, pp. 9–21.

McKenzie, Erica D et al. (2017). “Validation of a smartphone-based EEG among people with epilepsy: A prospective study”. In: *Scientific reports* 7, p. 45567.

McLane, Hannah C et al. (2015). “Availability, accessibility, and affordability of neurodiagnostic tests in 37 countries”. In: *Neurology* 85.18, pp. 1614–1622.

Computational Neuroscience, Swartz Center for (n.d.). *Makoto's preprocessing pipeline*. Last modified: 23 December 2020, Retrieved: 02 January 2021. URL: https://sccn.ucsd.edu/wiki/Makoto's_preprocessing_pipeline.

Neurotech, Novela and NeuroTechX (n.d.). *Neureka™ 2020 Epilepsy Challenge*. Retrieved November 16, 2020. URL: https://neureka-challenge.com/.

Chatzichristos, Christos et al. (2020). "Epileptic Seizure Detection in EEG via Fusion of Multi-View Attention-Gated U-net Deep Neural Networks". In: *Proceedings of the IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, p. 7.

Oostenveld, Robert and Peter Praamstra (2001). "The five percent electrode system for high-resolution EEG and ERP measurements". In: *Clinical neurophysiology* 112.4, pp. 713–719.

Organization, World Health et al. (2004). *Atlas: country resources for neurological disorders 2004: results of a collaborative study of the World Health Organization and the World Federation of Neurology*. World Health Organization.

Sokolov, Elisaveta et al. (2020). *Smartphone EEG Utility and Quality for Epilepsy Patients in the West African Republic of Guinea (196)*.

Berg, Anne T et al. (2010). "Revised terminology and concepts for organization of seizures and epilepsies: report of the ILAE Commission on Classification and Terminology, 2005–2009". In: *Epilepsia* 51.4, pp. 676–685.

Gramfort, Alexandre et al. (2014). "MNE software for processing MEG and EEG data". In: *Neuroimage* 86, pp. 446–460.

Wikipedia (n.d.). *International 10-20 system for eeg-mcn.svg*. Retrieved: 28 March 2021. URL: https://commons.wikimedia.org/wiki/File:International_10-20_system_for_EEG-MCN.svg.

Schindler, Kaspar et al. (2007). "Assessing seizure dynamics by analysing the correlation structure of multichannel intracranial EEG". In: *Brain* 130.1, pp. 65–77.

Kim, D and S Keene (2019). "Fast automatic artifact annotator for EEG signals using deep learning". In: *2019 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*. IEEE, pp. 1–5.

Bergstra, James, Daniel Yamins, and David Cox (2013). "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures". In: *International conference on machine learning*. PMLR, pp. 115–123.

A Jiang survey: EEG artifact processing recommendations

Method	Additional Reference	Automatic	Online	Can Perform on Single Channel
Regression	Y	Y	N	N
Wavelet	N	Y	N	Y
ICA	N	N	Y	N
CCA	N	N	Y	N
Adaptive filter	Y	Y	Y	Y
Winner filter	N	Y	N	Y
Wavelet BSS	N	N	N	Y
EDM BSS	N	N	N	Y
BSS-SVM	N	Y	Y	N

Table A.1: Adapted from original source Jiang, Bian, and Tian 2019

Technical
University of
Denmark

Richard Petersens Plads, Building 321
2800 Kgs. Lyngby
Tlf. 4525 1700

www.compute.dtu.dk