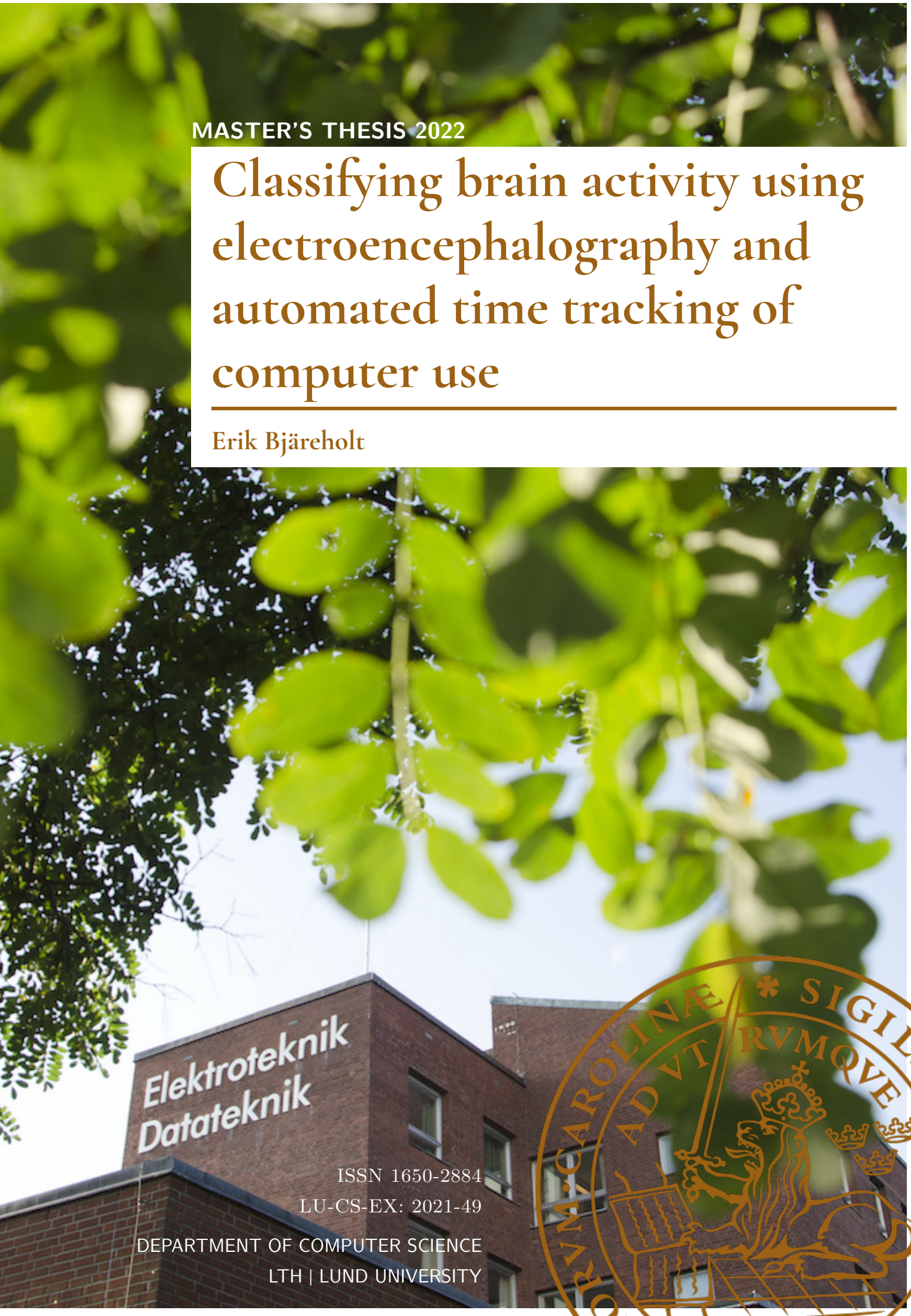


MASTER'S THESIS 2022

Classifying brain activity using electroencephalography and automated time tracking of computer use

Erik Bjäreholt

ISSN 1650-2884
LU-CS-EX: 2021-49

DEPARTMENT OF COMPUTER SCIENCE
LTH | LUND UNIVERSITY

EXAMENSARBETE

Datavetenskap

LU-CS-EX: 2021-49

Classifying brain activity using electroencephalography and automated time tracking of computer use

Klassificering av hjärnaktivitet med elektroencephalografi och automatiserad tidsspårning av datoranvändning

Erik Bjäreholt

Classifying brain activity using electroencephalography and automated time tracking of computer use

Erik Bjäreholt
erik@bjareho.lt

December 22, 2021

Master's thesis work carried out at RISE.

Supervisor: Markus Borg, markus.borg@{cs.lth.se, ri.se}

Examiner: Elizabeth Bjarnason, elizabeth.bjarnason@cs.lth.se

Abstract

We investigate the ability of EEG to distinguish between different activities users engage in on their devices, building on previous research which showed a considerable difference in brain activity between code- and prose-comprehension, as well as differences during code- and prose-synthesis. We perform a replication study and improve upon past results using state-of-the-art machine learning classifiers based on Riemannian geometry.

Furthermore, we extend the scope of previous work by introducing the automated time tracking application ActivityWatch, to track the device activities that the user is engaging in. This lets us label EEG data with naturalistic device activity, which we then use to train classifiers to discern activities such as code writing vs prose writing, or work vs media consumption. Our results indicate that a consumer-grade EEG device can discern between different activities that a user performs at the computer. Among other results, we show that not only can code and prose *comprehension* be distinguished, but also code and prose *writing*.

A full replication package, including source code and a sample dataset, is available at github.com/ErikBjare/thesis

Keywords: electroencephalography, brain computer interfaces, time tracking, code comprehension, productivity, MSc

Acknowledgements

- My advisor Markus Borg.
- My brother Johan Bjäreholt and all the ActivityWatch contributors, for working with me all these years.
- The NeuroTechX crowd, specifically John Griffiths and Morgan Hough, for their support and time spent helping me.
- Pex Tufvesson and Carolina Bergeling at the Department for Automatic Control, for providing early guidance.
- Andrew Jay Keller at Neurosity, for gifting me a refurbished Notion DK1 to work with.
- Alex K. Chen, for referring me to all the right people.
- All the test subjects, for their time and interest.
- Everyone who has contributed to the open source tools I have used.
- Everyone who have supported me at LTH.
- Friends and family, for their neverending love and support.

Contents

Preface

When I started university in 2013, I already knew that one day I wanted to work with brain-computer interfaces. During high school I had been soaked in transhumanist and hacker culture, spending more time online than in school.

I had come to the realization that the field was bound to fundamentally change how we interact both with computers and each other. However, I was aware that the field was in its earlier phases, and the work that needed to be done was outside of my expertise at the time.

So I asked how can I set myself up to be in a position to contribute to this impactful field of research. My answer at the time was to learn about data collection and analysis generally, and a couple years later applied my skills building ActivityWatch, as I figured the best approximation about what goes on in my head is what I'm looking at on my computer screen.

What I thought would be a small learning project soon took on a life of its own. People online started showing interest, and soon my brother joined in development. What started as a fun project to analyze my own device usage soon became a popular open source application with thousands of users. At some point it became apparent that once my Master's thesis was to be written, it would involve ActivityWatch one way or another.

As I was finishing up my final exams, I found an old paper in a moving box, it was a thesis proposal I had picked up at a career fair many years back. The proposal was from my advisor Markus Borg, and curious as I was, I contacted him and asked if it was still available. Markus enthusiastically replied, and helped me adapt his suggestion to combine it with my previous work on ActivityWatch.

As my thesis work progressed, I learned a lot about brain imaging and BCIs in general, and EEG in particular.

This thesis work has served as my first proper contribution to the field, all according to the plan I had made at the start of my university degree.

The Oxford English Dictionary defines 'thesis' as "a long essay or dissertation involving *personal research*, written by a candidate for a university degree". I can not think of more "personal research" than research in quantified self with personal data.

Chapter 1

Introduction

The rise of computing over the last half century has led to an unprecedented rise in new methods for data acquisition and analysis. However, our ability to quantify and analyse brain function is still limited. With advanced and expensive equipment, such as *Magnetic Resonance Imaging* (MRI) scanners and high-density *Electroencephalography* (EEG) headsets, we can record brain activity at an adequate resolution to draw conclusions about neural activation patterns under different behaviours.

While this professional-grade equipment is still unavailable to the general public, in recent years some brain imaging techniques have become commercialized for various purposes (such as meditation aids, biofeedback) and sold as consumer devices [1]. Consequently, the cost of EEG devices has fallen sharply, offering an affordable solution to monitoring brain activity.

Although many different tasks have already been studied with EEG, each task to be studied requires a time-consuming controlled experiment. To increase the diversity of tasks under study without needing to create task-specific experiments, we introduce our previously developed time-tracking application ActivityWatch [2] to track and categorize the computer activities a subject engages in during natural device use. Using ActivityWatch and EEG we can therefore collect both device use data and brain activity data, letting us study brain activity during particular device activities.

Studying brain activity during computer-based tasks is of particular interest to software engineers and knowledge workers. For example, research has been done on:

- Studying differences in code- and prose-comprehension [3, 4].
- Measuring the difficulty of a development task [5].
- Minimizing interruptions [6].
- Recognizing emotions during programming [7].

In addition to academic research, some communities like the Quantified Self movement have risen to the challenge of collecting and analyzing personal data for insights into our lives and health [8]. But while some data is easily observed (such as device usage, location, and physical movement), it remains difficult to collect data about subjective attributes (like mood, focus) [9]. To collect data on these more elusive phenomena, researchers and individuals have to resort to manual data collection like questionnaires, or approximations constructed from other data, leading to significant time costs and scaling difficulties [9]. This makes low-cost brain imaging techniques like EEG a budding new frontier for the Quantified Self movement in particular, and citizen science in general.

In this thesis we have developed a framework and tooling for studying brain activity during uncontrolled device use (*naturalistic device activities*). Using our previously developed time-tracking application ActivityWatch [2], we can track the subject's device activity while simultaneously collecting their brain activity data using consumer-grade EEG devices. We then use the EEG and device usage data to train machine learning models to discern which task the subject is engaged in.

We also apply similar methodology to replicate a controlled experiment reported by Fucci et al. in 2019 [4], where we present a small number of subjects with a set of code- and prose-comprehension tasks while being monitored with EEG. We use the collected data to train machine learning classifiers, evaluating the feasibility of low-cost EEG equipment to distinguish between the code and prose tasks, and use the results as an indication whether it could extend to other tasks, as in our naturalistic setting experiment.

Our results confirm the findings of the original study, i.e., EEG data can be used to distinguish between code and prose-comprehension. Furthermore, the classifier based on Riemannian geometry outperforms the bandpass-features used by Fucci et al. in terms of classification accuracy.

In our naturalistic experiments, our results indicate that a consumer-grade EEG device can discern between different activities that a user performs at the computer. Among other results, we show that not only can code and prose *comprehension* be distinguished, but also code and prose *writing*.

1.1 Functional brain imaging

Functional brain imaging methods such as *Functional Magnetic Resonance Imaging* (fMRI), *Functional Near-Infrared Spectroscopy* (fNIRS), and EEG, have been used to study the relationship between cognitive or physical activity, and brain activity [3, 10, 4]. The more accurate methods, such as fMRI, are costly and inflexible/impractical for many uses. The different methods in general come with a set of trade-offs. A common obstacle is cost: operating an fMRI machine costs on the order of $300/h [4].

Other considerations when picking a brain imaging method is temporal/spatial resolution. For instance, fMRI can measure activity deep in the brain (good spatial resolution), but has difficulty tracking activity over time due to relying on blood oxygenation, which is a much slower process than the underlying neural processes (bad temporal resolution) [11]. EEG on the other hand, has excellent temporal resolution (usually sampled at about 256Hz), but it can not reliably discern activity deep in the brain (bad spatial resolution, partially depending on channel count).

Recently, the availability and cost reduction of biosensors such as EEG, *Hemoencephalography* (HEG), and fNIRS, has enabled studying brain activity during real-life tasks. This is the opportunity we seek to take advantage of, and as EEG is the most well developed of the three, we choose it for our investigation.

EEG works by measuring tiny amounts of electricity (on the order of microvolts) on the scalp to listen to the underlying firing of neurons. By placing electrodes in various configurations, information about the activity of different brain regions can be inferred. The extent of how much information can actually be decoded from the signal remains an open research question.

But EEG is not without its limitations — among them a notably low signal-to-noise ratio [12], as well as being difficult to use for discerning activity deeper in the brain [13] — yet these limitations have been overcome for many applications, like *Event-Related Potential* (ERP) experiments, and has even proven sufficient for high-speed *Brain-Computer Interface* (BCI) applications through detecting *Visual Evoked Potentials* (VEPs) [14].

To combat the low signal-to-noise ratio, machine learning methods have been employed with varying degrees of success. Examples from previous research include *Convolutional Neural Networks* (CNNs), which have been successful in classifying time series in general [15], and EEG data in particular [16]. As well as *Hierarchical Convolutional Neural Networks* (HCNNs), which have been used for EEG-based emotion recognition [17].

Cost reduction

The cost-reduction of EEG devices took hold in 2013, when OpenBCI ran the founding crowdfunding round on Kickstarter. During the campaign, they offered their 16-channel Cyton + Daisy board (seen in Figure 3.3 on page 33) for $549[1] [18].

The commercialization of EEG towards a general audience was furthered by InteraXon with the release of the original Muse in 2016. Aimed at meditation practice, it was the first consumer-oriented EEG device on the market. Later, in 2019, InteraXon released the Muse S (seen in Figure 3.2 on page 32), a headband-style EEG headset developed for sleep and longer sessions.

In 2020 and 2021, the company Neurosity has released their 8 channel Crown headset (seen in Figure 3.4 on page 33), targeted at developers and knowledge workers to 'shift into focus'. Looking forward, projects like the FreeEEG32 offer a 32 channel board for only $199, expected to ship in 2022 [19].

In our experiments we have chosen the Muse as our primary device, but our code also supports OpenBCI-devices, the Neurosity Crown, and other devices compatible with the *brainflow* library.

Applications in neurolinguistics

Neurolinguistics is the study of how language is comprehended and produced by the brain. Within the field, many brain-imaging methods have found use to study both the where and how of language processing. The techniques chosen often depend on if researchers are asking question about *where* processing happens, requiring good spatial resolution, versus *how* it happens, requiring good temporal resolution (as described in Section 1.1 on the previous page).

EEG, due to its great temporal resolution, has found many applications in neurolinguistics, both to understand how the brain processes natural languages as well as programming languages [20].

As an example it has been shown that it is possible to classify if a participant is reading code or prose using fMRI [3], which has been replicated using EEG and low-cost biosensors [4]. The ability to distinguish between these two tasks have been has been explained neurologically by the recruitment of different neural networks in the brain [21].

The code vs prose comprehension task has also been modified into a writing task studied under fMRI, to further shed light to the underlying brain activity. In one study it was found that code and prose writing are significantly dissimilar tasks for the brain, where prose writing engages brain regions associated with language in the left hemisphere while while code writ-

[1] Fun fact: I funded them at one of the lower tiers back then.

ing engages brain regions associated with attention, working memory, planning, and spatial cognition in the right hemisphere [22]. To study this with MRI, the authors had to use an MRI-safe keyboard, as the powerful magnetic field prevents the use of normal electronics.

As far as I know, code vs prose *writing* has not been studied with EEG. However, using ActivityWatch and other tools we have developed in this thesis (like the input watcher) one can to distinguish whether the user is reading or writing, as well as whether they are working with code or prose, which enables studying the same activities in a naturalistic setting.

1.2 Automated time trackers

People spend more time than ever using computing devices. Work, entertainment, and services, have been steadily moving online over the last few decades and this trend is expected to continue. While several studies have been tracking how people spend time on their devices, a wider study of how people's app usage is changing over time, and how it varies with demographics, is not publicly available.

Furthermore, how different device activities affect the user behaviorally and neurologically is of interest for many areas of research, including:

- psychological well-being, such as depression and social anxiety [23, 24], stress [25], self-esteem, life satisfaction, loneliness, and depression [26].
- the impact of screen time on children and adolescents [27].
- attention span among media multitasking adults [25].
- enhancing personal productivity [28].

To study the effects of device use, much of the older literature uses surveys as their primary means of data collection. However, these surveys are time-consuming and prone to self-reporting bias. To address these issues, more recent research has employed the use of automated time-trackers to gain detailed and objective data about subjects' device activities.

These automated time-trackers have been developed for many types of devices (computers, phones, TVs), with various applications such as tracking hours worked (and on what), personal productivity, lifelogging, and managing excessive use of social media.

Understanding device use and the underlying cognitive processes are essential when designing for motivation, engagement and well-being in digital experiences [29]. As such, studying device activity together with brain activity seems a promising endeavor.

This section introduces how such devices are used in commercial services and in research. Finally, we present our own automated time-tracker ActivityWatch, used in this thesis.

Commercial use

Companies like RescueTime [30], Hubstaff [31], and others offer automated time tracking as a service. These services track users' screen time using an application which tracks the active application, and sends the data to their servers for storage and analysis. The user can then view their data in a dashboard on the services' website. Many of these services are marketed towards professionals who want to keep track of individual and team productivity.

However, these services have some issues for use by researchers and individuals alike. Notably, their collection of detailed and non-anonymized behavioral data into a centralized system bring significant privacy concerns, especially in cases where the data is shared with a team or an employer.

Other limitations of these services, such as low temporal resolution and limited event detail, cause additional issues for certain tasks that are timing-sensitive (such as ERPs), or preprocessing steps that can take advantage of high level of detail (like classifying activity).

Research use

Previous research has been published which used automated time trackers, such as TimeAware [28] (studying the impact of positive/negative nudging on device use) and ScreenLife [32] (studying users' interest in time tracking). However, these previous contributions are — like the commercial services — not open source nor permissively licensed, and therefore not available for external research use nor further development.

ActivityWatch

The free and open source automated time-tracker ActivityWatch [2], previously developed by the author of this thesis together with contributors. It addresses all the aforementioned issues around source availability/licensing, privacy, temporal resolution, event detail, and cross-platform support.

ActivityWatch works by tracking the active application, its window title, and whether the user is active or not (explained in Section 3.2.2 on page 37). That data is then analyzed to show how much time was spent on different applications, websites, or projects (as shown in Figure 1.1 on the next page). The project was started in 2016 by the author of this thesis, with his brother Johan joining development soon after. It has since become a popular open source alternative to other time tracking software. It has received numerous contributions from users[2] and has over 100,000 downloads[3].

[2]Contributor statistics available at activitywatch.net/contributors/
[3]Download statistics available at activitywatch.net/stats/

Figure 1.1: A screenshot of the ActivityWatch activity dashboard. Showing top categories, top window titles, an hour-by-hour stacked barchart, top applications, the category tree, and a sunburst visualization of the time spent in each category.

1.3 Aim of the thesis

When I started the thesis, we set a goal of answering two research questions:

- RQ1. Can we improve upon previous results in classifying code vs prose comprehension with EEG?
- RQ2. Can we train an EEG classifier to separate software developers' device activities?

The primary aim of the thesis, as described in RQ1, is to improve upon previous attempts [4] to classify whether the user is reading code or prose using EEG data. This is to be achieved by using better EEG equipment and explore state of the art analysis methods, such as Riemannian geometry and neural networks.

The secondary aim of the thesis, as described in RQ2, is to investigate if we can generalize the results from classifying code vs prose comprehension to a wider variety of tasks engaged in during natural device use. This is to be done by using ActivityWatch to label the device activities, and then train classifiers to discern the different activities.

Additional contributions of the thesis include:

- Developing a method for automatically labelling time-series data, such as EEG data, using ActivityWatch
- Providing a complete reproduction package, including all code and data.
- Improving open-source tools for EEG acquisition and analysis.

As part of the work on this thesis I have contributed several improvements to some of the open source projects used, including muse-lsl, moabb, eeg-notebooks, pyriemann, and braindecode.

- Pull requests to NeuroTechX repos (eeg-notebooks, moabb) (22)
- Pull requests to muse-lsl (3)
- Pull requests to pyriemann (1)
- Pull requests to braindecode (1)

1.4 Related work

Previous research shows that fMRI (by Floyd et al. [3]) and EEG (by Fucci et al. [4]) can be used to classify whether a subject is reading prose or code. However, accuracy with single-channel EEG has been found to be poor, and notably outperformed by a heart rate variability (HRV) monitor. This is a result we seek to improve upon by using better EEG equipment.

One paper in particular utilized other low-cost biosensors (such as wristbands that measure electrodermal activity and heart-related metrics) to perform emotion detection on developers at work, as a tool to study developer productivity [7]. In our review of the literature we find emotion detection to be a hard problem, with considerable difficulty in designing up robust experiments that reliably elicit the desired emotions in subjects. While we initially also considered attempting emotion classification, we ultimately chose not to due to the aforementioned issues around experimental design.

Returning to our code vs prose task, it has recently been shown that the multiple demand (MD) system[4] is typically recruited for code comprehension tasks, as opposed to the language system[5] that is typically recruited during prose comprehension [21]. This sheds light on the significant differences in how the brain processes code vs prose.

In addition to purely studying comprehension (reading) code and prose, an 2020 fMRI study showed that there are indeed significant neurological differences between *writing* code and prose as well [22]. This indicates that code and prose are dissimilar not only in *comprehension* but also in *synthesis*, as we also investigate in our naturalistic device use experiments.

A systematic mapping study has also been conducted by Vieira et al. [33] which reviews the usage of psychophysiological data in software engineering. They filtered out 27 studies and found that the most used technique was EEG (used in 8 out of 27 studies), followed by eye tracking and fMRI (each used in 5 studies). They found that two thirds of the studies aimed to understand how developers understand source code and which strategies they use during debugging. They observe a greater number of publications in recent years, and highlight the integration between neuroscience and software engineering as a way to open up for innovative studies that make use of psychophysiology in the software development process.

[4] A set of frontoparietal brain regions involved in diverse tasks

[5] Specialized brain regions that respond selectively during language processing

Chapter 2

Theory

This chapter serves to introduce relevant theory on which the thesis rests. We will briefly review EEG, its use in research and practice. We also briefly review machine learning techniques used on EEG data, in particular Riemannian geometry.

2.1 Electroencephalography (EEG)

Electroencephalography is a method used to measure the activity of neurons in the brain by recording the electrical potential of the scalp. These measurements are performed (sampled) hundreds of times per second, and are then put in sequence to form a signal of what can informally be referred to as "brain waves".

As a non-invasive method it is widely used in medicine to diagnose and study a wide range of conditions, including epilepsy and sleep disorders. It has also found use in research, in particular for studying *Event-Related Potentials* (ERPs), which are stereotyped responses to a stimulus, an example of which can be seen in Figure 2.1. Examples of common ERPs can be seen in Table 2.1.

One difficulty of working with EEG lies in the low signal-to-noise ratio: measurements are on the order of microvolts, and what is being measured is the electrical activity of some subset of the approximately 86 billion neurons in the human brain. This often leads to signal artifacts from other, non-neural, electrical activity near the scalp, such as that caused by muscle activity from e.g. eye blinks (as seen in Figure 2.6 on page 25).

ERP	Elicited by
N170	Processing of faces, familiar objects or words.
N400	Words and other meaningful stimuli.
P300	Decision making, oddball paradigm.
P600	Hearing or reading grammatical errors and other syntactic anomalies.

Table 2.1: Common *Event-Related Potentials* (ERPs)

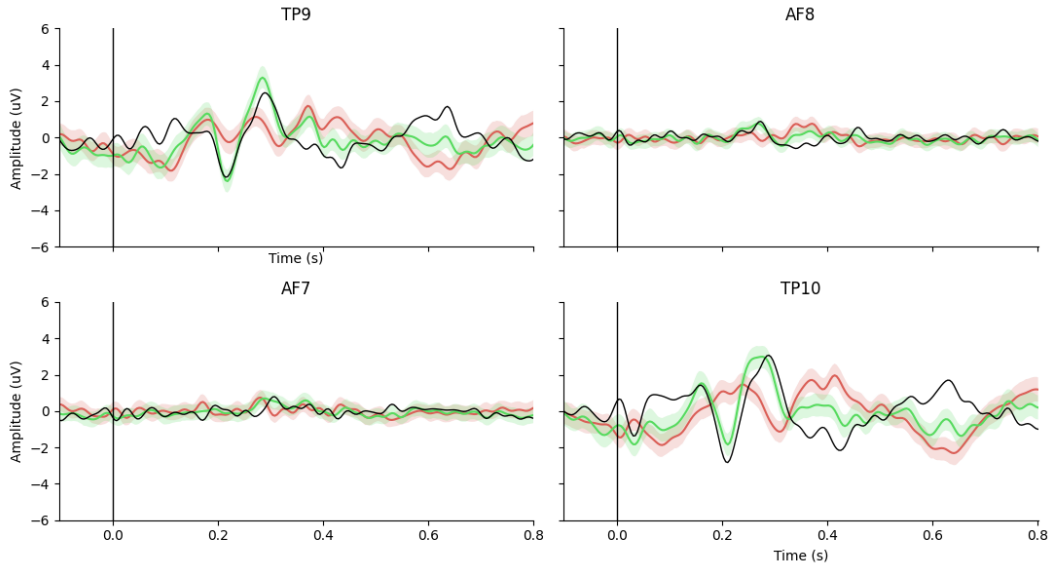


Figure 2.1: Example of aggregated epoch-averaged ERP trials from an N170 experiment, where the subject was shown either a face (in **green**) or a house (in **red**). The difference between the two stereotyped responses can be seen in black. The vertical line marks the beginning of the epoch.

Source: *eeg-notebooks experiment gallery (BSD 3-clause)*

The non-invasive nature of EEG can be contrasted with *Electrocorticography* (ECoG), a form of *Intracranial Electroencephalography* (iEEG), which is an invasive method where a craniotomy (a surgical incision on the skull) is performed to implant an electrode grid directly on the cerebral cortex. The result is higher spatial resolution, and improved signal-to-noise ratio due to closer proximity to neural activity. It is almost exclusively used in patients with intractable epilepsy (not responding to anticonvulsants).

Electrodes can be placed at different locations on the scalp, targeting different regions of the brain. The system used to position the electrodes is called the 10–20 system (seen in Figure 2.2), and is the standard way to label electrode placements. The *10* and *20* come from the fact that distances between adjacent electrodes are either 10% or 20% of the total front–back or right–left distance of the scalp.

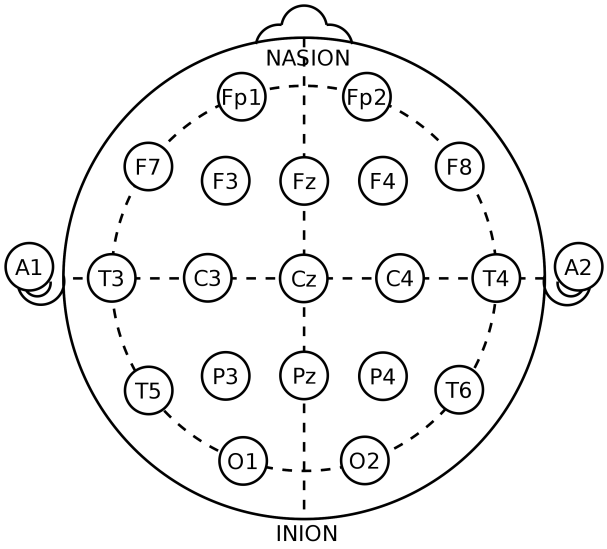


Figure 2.2: The 10–20 system.

Source: *Wikimedia Commons (public domain)*

For configurations with high-density or more specific electrode placement there is also the extended 10–10 system, using MCN. In this system, the electrode placements are divided by 10% increments instead of the usual 20% (seen in Figure 2.3).

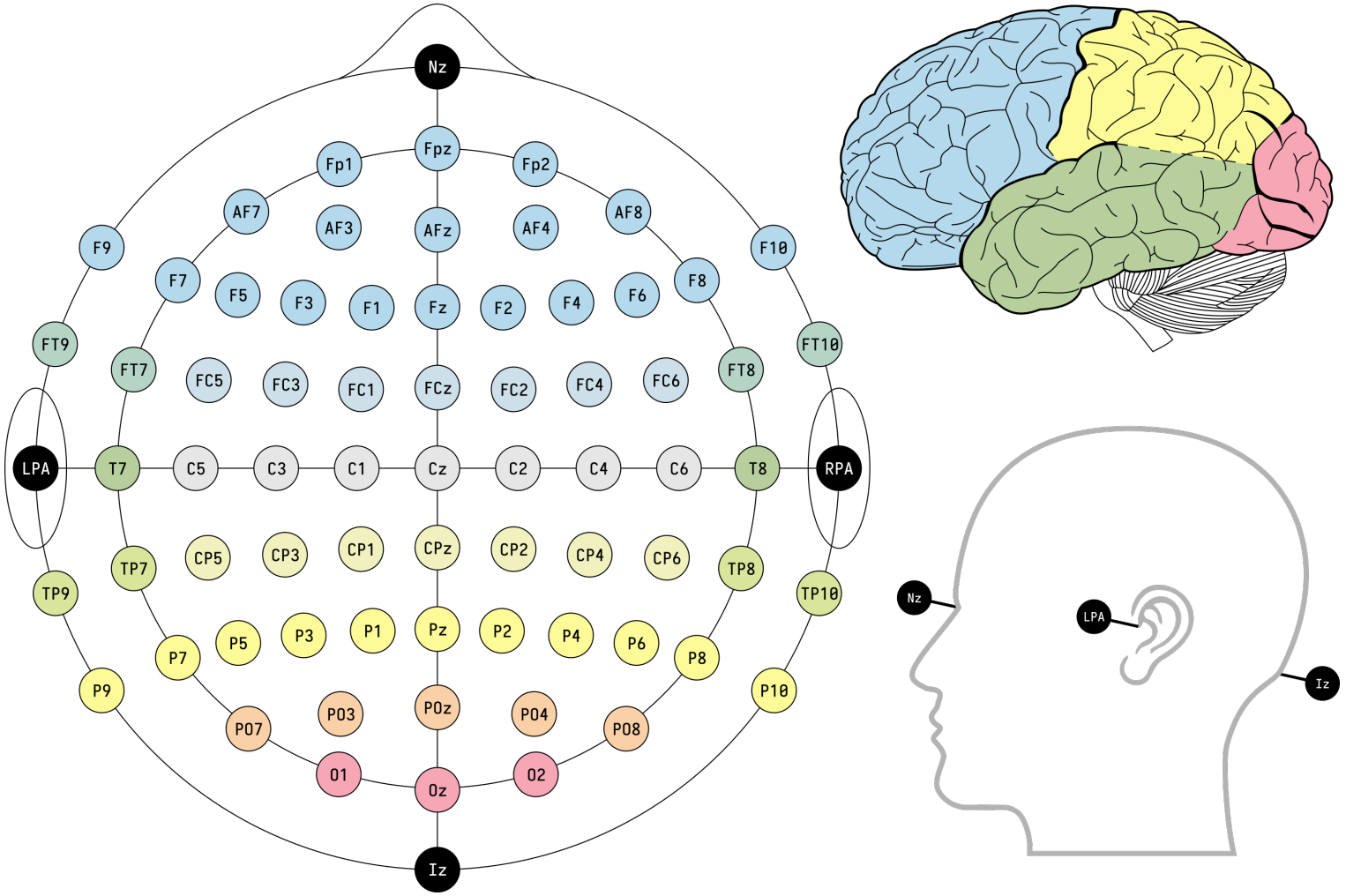


Figure 2.3: The extended 10–10 system, following the Modified Combinatorial Nomenclature (MCN).

Source: *Wikimedia Commons (CC-0)*

As measurements are taken on the scalp, neural activity of surface-level neurons (such as the cerebral cortex) are expected to dominate the signal, which makes it difficult to use when studying systems deeper in the brain (such as the hippocampus). As an example of this, eye blinks are easily identifiable in the signal when electrodes are placed on the frontal cortex (seen in Figure 2.6).

Raw EEG data usually contain artifacts from powerline noise, which needs to be filtered out. This is usually done with a simple bandpass filter, an example of which can be seen in Figure 2.4 and 2.5.

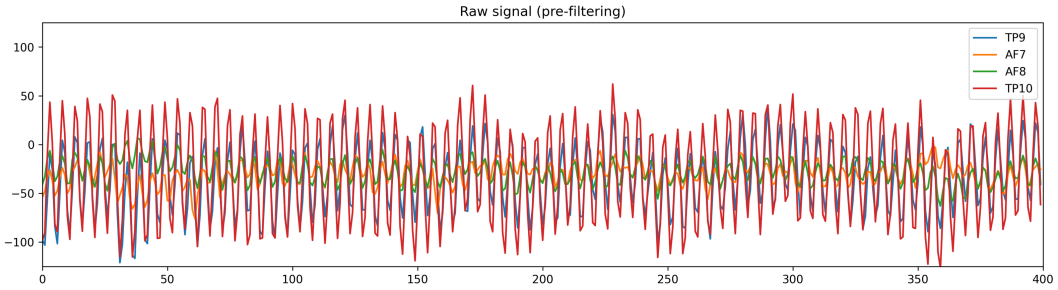


Figure 2.4: Raw EEG signal with no filtering. Powerline noise clearly visible.

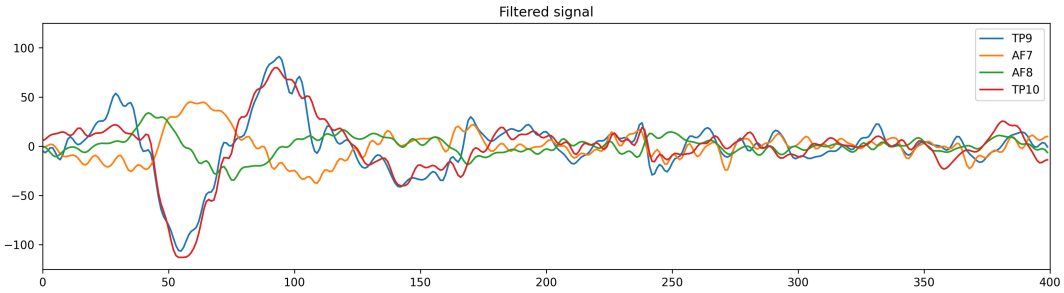


Figure 2.5: Same EEG signal as in Figure 2.4, but with bandpass filter applied. We can see that powerline noise has been eliminated.

After the signal has been filtered, one can perform spectral density approximation to estimate the relative power of different frequency bands in the signal. An example of frequency bands commonly used can be seen in Table 2.2. We will come back to this later in the method, where we use this information as features for one of our classifiers.

Frequency band	Frequencies	Brain states
Gamma (γ)	>35 Hz	Concentration
Beta (β)	16–35 Hz	Active, external attention, relaxed
Sigma (σ)	12–16 Hz	Sleep spindles
Alpha (α)	8–12 Hz	Very relaxed, passive attention
Theta (θ)	4–8 Hz	Deeply relaxed
Delta (δ)	0.5–4 Hz	Sleep

Table 2.2: Characteristics of common frequency bands used in EEG research. Note the *Sigma* band, which is usually grouped with the *Beta* band, but is often used in sleep research where it targets sleep spindles during stage 2 NREM sleep. In some research, these bands are split further into slow and fast counterparts.

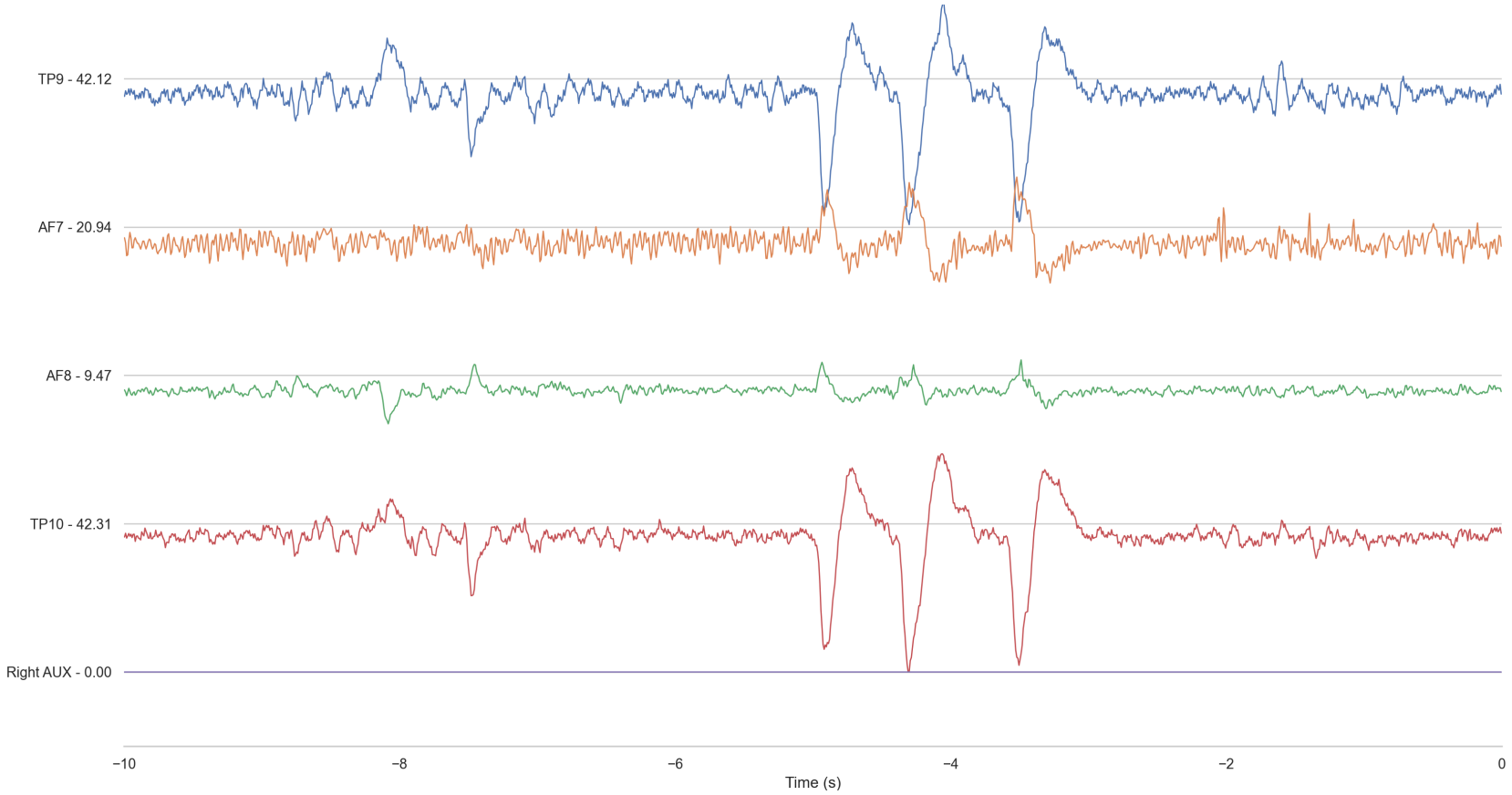


Figure 2.6: EEG signal viewed with muse-lsl. There are 3 eye blinks identifiable around **−4** s. The signal has been bandpass filtered to eliminate noise.

2.2 Machine Learning

Machine learning on EEG data utilizes several domain-specific methods, often similar to other methods for time-series data in general. However, some methods differ by taking advantage of the spatial information that is available through the analysis of interdependence between multiple EEG channels [34].

Methods used in analysis and classification of EEG data include Linear Discriminant Analysis (LDA), Common Spatial Pattern (CSP) filters [35], spectral density estimation (bandpower features), and the computation of covariance matrices to estimate interdependencies between channels. The underlying ML algorithms being trained on the features are often off-the-shelf logistic regression, support vector machines, random forests, etc. Some domain adaptations are found in more complex models like neural networks.

Furthermore, preprocessing methods include bandpass filtering, windowing, and various tricks for channel selection (in high-density setups). With the use of these methods, we can preprocess the data, compute features, and train our classifiers.

In our study, we will primarily be using Riemannian methods, and compare them to a common bandpower-features approach (used by Fucci et al.) as our baseline.

2.2.1 Riemannian geometry

EEG decoding approaches based on Riemannian geometry have led to state-of-the-art classification performance on many tasks [34]. The idea behind it is to take advantage of the spatial structure of EEG, i.e. the fact that there are multiple channels which co-vary in specific ways.

The Riemannian distance metric δ_G for two symmetric positive definite matrices $\mathbf{A}$ and $\mathbf{B}$ (such as covariance matrices) is [36]:

$$\delta_G(\mathbf{A}, \mathbf{B}) = \sqrt{\sum_{i=1}^{N} \ln^2 \lambda_i(\mathbf{A}, \mathbf{B})}$$

Where $\lambda_i(\mathbf{A}, \mathbf{B})$ are the eigenvalues from $|\lambda\mathbf{A} - \mathbf{B}| = 0$.

The naive Riemannian approach using *Minimum Distance to Mean* (MDM) starts with computing the covariance matrix for each trial, and then estimating a mean covariance matrix for each class (the class centroid) using the Riemannian distance metric δ_G. When classification is performed, the distance between the new covariance matrix and the class centroids is estimated using the Riemannian distance metric, and the new covariance matrix is classified according to which class centroid is closest.

To allow for use of other classification approaches in the final step, such as logistic regres-

sion and SVMs, we can project our covariance matrices onto a tangent space (a schematic representation of this can be seen in Figure 2.7). Once projected onto this tangent space, our covariance matrices can be compared fairly well using the standard Euclidean distance metric, letting us use common methods which do not employ the Riemannian distance metric [37].

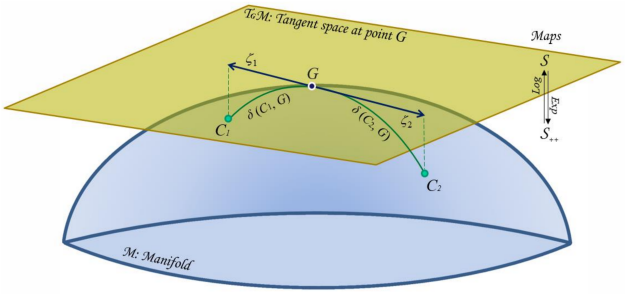


Figure 2.7: Schematic representation of the symmetric positive definite matrix manifold, the geometric mean $\boldsymbol{G}$ of two points and the tangent space at $\boldsymbol{G}$. The geometric mean of these points is the midpoint on the geodesic connecting $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, i.e. it minimizes the sum of the two squared distances. The map from the tangent space to the manifold is an exponential map. The inverse map is a logarithmic map.

Source: *Congedo et al. [37]*

Chapter 3

Method

We start by deciding on an EEG device to use, we then collect data for both the controlled and naturalistic experiments. For the controlled experiment (RQ1), we collect our labels from the presented stimulus. For the uncontrolled experiment (RQ2), we collect device activity labels during natural device use using ActivityWatch. Once the data has been collected, we perform data labelling, transformation, and cleaning, before we finally train our classifiers.

Our main classifier is based on Riemannian geometry. For our controlled experiment, we also train a classifier based on bandpower-features as a baseline reference. We evaluate the performance of our classifiers using the balanced accuracy metric and LORO cross-validation.

An overview of the research method is presented in Figure 3.1. The analysis step is further broken down in Figure 3.8 on page 38.

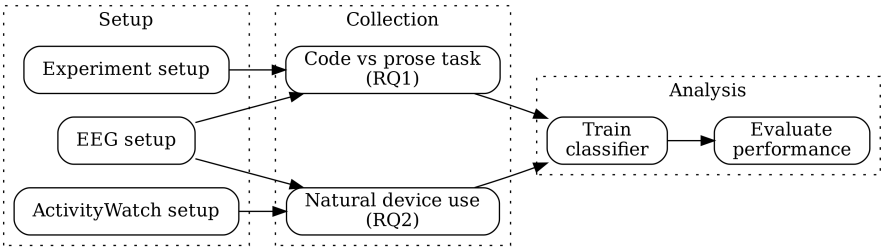


Figure 3.1: Overview of the research method for the two tasks: controlled code vs prose experiment, and naturalistic use.

The method for our controlled experiment mimics Fucci et al. [4] with some differences (seen in Table 3.1 on the facing page):

- We use EEG exclusively.
- We used a different device.
- We have a smaller & less diverse sample.
- We have reimplemented the task in eeg-notebooks.
- We have used the original prose-review stimuli by Floyd et al. [3].

Some of these differences are motivated and discussed further later in the report, especially in Section 5.3 on page 50.

Setting	Study		
	This study	**Fucci et al.** (2019)	**Floyd et al.** (2017)
Experiment site	Lund Univ. (Sweden)	Univ. of Bari (Italy)	Univ. of Virginia (USA)
# Participants	10	28	29
Participants experience	Grads	Undergrads	Grads & Undergrads
# Tasks	Variable	36 tasks	27 tasks
Task type	Code comprehension Prose review	Code comprehension Prose comprehension	Code comprehension Code review Prose review
Physiological signal	Neural	Neural Skin Heart	Neural
Physiological measure	EEG	EEG EDA HR, HRV, BVP	BOLD
Device	Muse S	BrainLink Headset Empatica wristband	fMRI
Classifier	Riemannian geometry	8 algorithms	Gaussian Process
Classifier validation	LORO-CV	LORO-CV Hold-out	LORO-CV
Classifier metric	Balanced accuracy (BAC)	Balanced accuracy (BAC)	Balanced accuracy (BAC)

Table 3.1: Comparison of this study's method with previous studies.

3.1 Devices

In this section we will briefly describe the different EEG devices we have worked with, and their differences.

We experimented with several devices but eventually settled on the Muse S for both our experiments. The motivation for choosing the Muse S was mainly due to comfort and ease of use. The other devices considered included the OpenBCI Cyton (with the Ultracortex headset), and the Neurosity Crown.[1]

In our framework, we have implemented support all of the headsets mentioned (and more), such that future work can use whichever device is desired.

Manufacturer	Device	Channels	Sampling rate	Comfort
InteraXon	Muse S (2020)	4	256Hz	High
Neurosity	Crown (2021)	8	256Hz	Medium
OpenBCI	Cyton (2013) + Ultracortex	8–16	125–250Hz	Low

Table 3.2: Devices used

Muse S

The Muse S is a 4-channel EEG headband with electrodes at TP9, AF7, AF8, and TP10 (using the modified 10–10 system, seen in Figure 2.3 on page 22), with the reference electrode at Fpz [38]. Its main advantage is its very comfortable form-factor and long battery life. It's limited by its channel count and electrode placement. It connects using Bluetooth.

Figure 3.2: Muse S

[1] Earlier in the work, before we received the Crown, we were also generously gifted a Neurosity Notion DK1 to get a head start.

OpenBCI Cyton

The OpenBCI Cyton is a 8-channel EEG board. It is commonly used with the partly 3D-printable OpenBCI Ultracortex Mark IV headset, which allows for flexible electrode placement. OpenBCI also offers an expansion board, the Daisy, which adds 8 additional channels. It connects using Bluetooth.

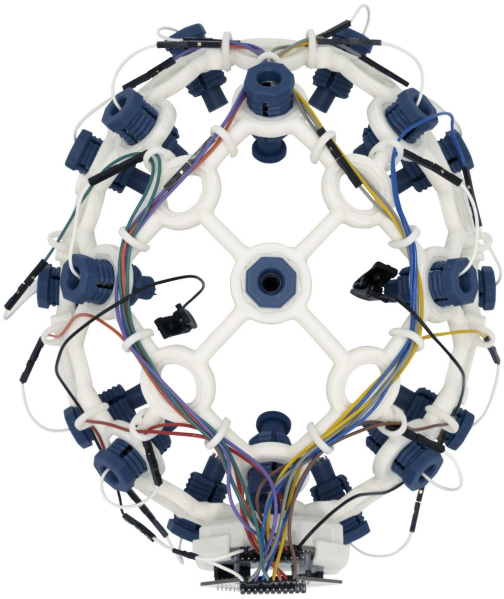


Figure 3.3: OpenBCI Cyton + Daisy with Ultracortex Mark IV

Neurosity Crown

The Neurosity Crown is a 8-channel EEG headset. It runs Linux on a quad-core CPU and 1GB RAM, and connects via WiFi. It has electrodes placed at CP3, C3, F5, PO3, PO4, F6, C4, CP4. Reference and bias electrodes at T7 and T8.

Figure 3.4: Neurosity Crown

3.2 Data collection

We need two types of data for the experiments: EEG data, and labels. For the controlled experiment, our labels are whether the stimulus is code or prose. For the naturalistic use experiment, our labels are the device usage categories.

To collect data for the controlled experiment, we:

1. Set up the EEG equipment
2. Use eeg-notebooks to present the stimulus, and record the time markers of when the stimulus changed as labels.

To collect data for the naturalistic use experiment, we:

1. Set up the EEG equipment
2. Configure ActivityWatch to collect device usage data

3.2.1 Collection of EEG data

EEG data was collected for the controlled experiment and during naturalistic device use. For both conditions, code from the open source `eeg-notebooks` [39] was adapted to record the raw EEG stream into a CSV file.

Depending on the device used we require certain software to connect to the devices. We used muse-lsl for the Muse S [40] which in turn uses Lab Streaming Layer. To support OpenBCI and Neurosity devices we used brainflow [41].

We have developed 'brainwatch', a command-line interface tool, to help us connect to and record from EEG devices. To connect & record from an EEG device, we can simply run:

```
$ brainwatch connect --device museS

Searching for Muses, this may take up to 10 seconds...
Found device MuseS-2754, MAC Address 00:55:DA:B9:27:54
Recording to recording_2021-09-29-09.22.24.csv
```

We can then check signal quality with `brainwatch check`, or run `brainwatch plot` to plot the raw signal.

Alternatively, we can plot using muse-lsl's built-in plotting functionality: `muselsl view -b Qt5Agg` (seen in Figure 2.6).

During naturalistic device use

For the naturalistic device use conditions, we used the Muse S due the superior comfort and ease of use compared with the alternatives, making it especially suitable for long recordings.[2]

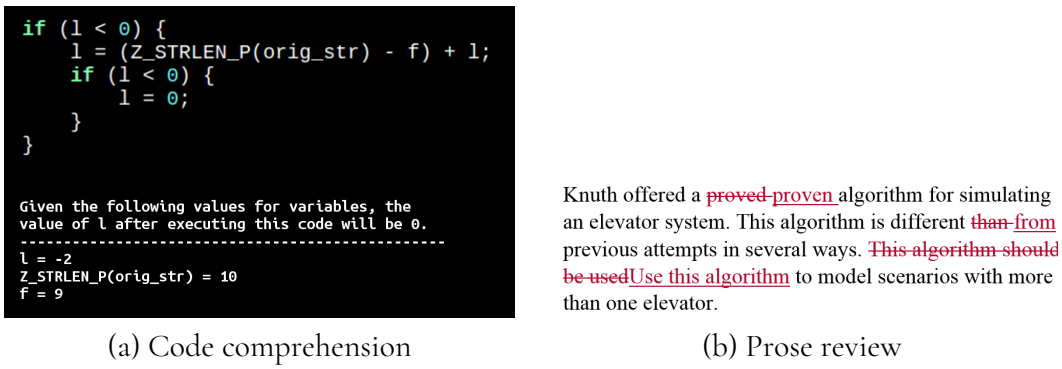
For this experiment we use a single-subject (the experimenter). The subject was asked to go about their usual device activities, often consisting of a mix of work (writing code, writing prose, email) and leisure (watching YouTube, reading Twitter).

We collected approximately 5 hours of EEG data during natural device use, and use the classes defined in Section 3.2.2 as labels for the data.

During code vs prose comprehension task

For the controlled condition, we ended up using the Muse S as well due to the comfort and ease of setup. We have 9 subjects, sampled by convenience from software engineers. The subjects were 8 males and one female. Most are in their late 20s or early 30s, except two who are in their 40s.

The experiment consists of presenting images with code or prose comprehension tasks, seen in Figure 3.5. These images are from previous studies on code vs prose comprehension by Floyd et al. [3] and Fucci et al. [4].



(a) Code comprehension (b) Prose review

Figure 3.5: Sample of the tasks used as stimuli.

We note that Fucci et al. modified the prose comprehension images from the original review-like task to become more comprehension-like, arguably better suited for the task (seen in Figure 3.6). However, due to the modified images being in Italian, we have used the original prose review images by Floyd et al.

[2] A wet electrode cap system was also considered, but was not used due to being inconvenient to setup.

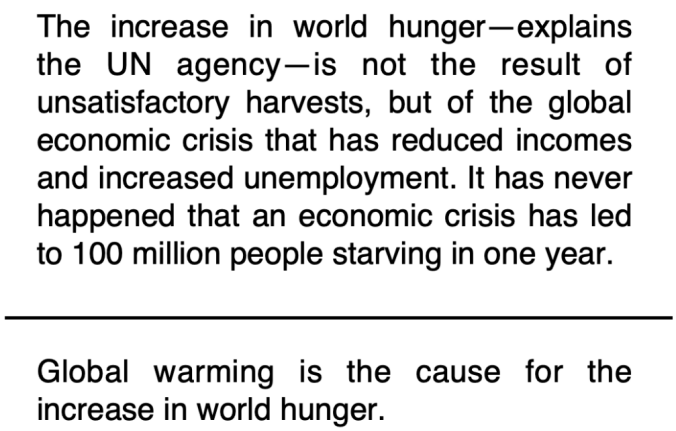


Figure 3.6: Sample of the prose comprehension task used by Fucci et al. (translated from Italian).

We implemented the task in eeg-notebooks [39], which uses previously mentioned libraries for data collection as well as PsychoPy [42] to present the stimulus.

Before each run, the subject was asked about their gender, age, handedness, and software development experience (specifically experience with C/C++). For good measure, we also asked if the subject had consumed caffeine the hours prior to the experiment.

After these questions we put the devices on and ensure we get good signal by inspecting it in real time with the viewer provided by muse-lsl (seen in Figure 2.6). The viewer itself does simple bandpass filtering between 3–40Hz, and the signal quality is indicated by the standard deviation of the filtered signal.

The stimulus images were presented in random order, for all subjects except subject #1 due to experimenter error (seen in Figure 4.1). For each stimulus the subject is asked to answer True/False to the questions posed by the code comprehension stimulus, or Correct/Incorrect for the prose review images. There is a short rest period of 5s between each stimulus. Each experiment run lasted between 10–25 minutes, depending on subject tiredness (subjects were asked to stop when tired).

3.2.2 Collection of device activity data

Device activity data is collected using our automated time-tracker ActivityWatch [2] (introduced in Section 1.2 on page 15).

ActivityWatch collects data through modules called watchers which report to the ActivityWatch server. It comes with two watchers by default:

- aw-watcher-window, tracks the active window and its title
- aw-watcher-afk, tracks if the user is active or not by observing input device activity

We have also built a custom watcher, aw-watcher-input, to track metrics of mouse and keyboard activity. It tracks by listening to mouse and keyboard events and records the distance in pixels the mouse moves and the number of key presses (but not which key was pressed). Every second this is bundled into an event, the values are reset, and then it continues with the next event. It was inspired by similar functionality in Andrej Karpathy's ulogme [43]. However, we did not have time to include it in analysis, and leave the integration of it for future work.

The data from ActivityWatch is processed and categorized such that the resulting data has the 3 columns `start, stop, class`. The class is determined by a regular expression that matches on window titles and URLs. For example, the regular expression `GitHub|github.com` could be used to match GitHub use.

We chose a few classes for analysis, seen in Table 3.3. Among them we have the classes *Programming* and *Writing*, to see if we can separate these activities in a natural setting.

Activity	Regular Expression
Programming	`[.](py|rs|js|ts)`
Writing	`[.](tex|md)`
Twitter	`Twitter|twitter.com`
YouTube	`YouTube|youtube.com`

Table 3.3: Activity classes used to label EEG data.

```
start,               stop,               class
2020-09-20 13:32:51, 2020-09-20 13:34:09, Twitter
2020-09-20 14:44:08, 2020-09-20 14:46:11, YouTube
2020-09-21 09:49:04, 2020-09-21 09:49:35, GitHub->Issues
2020-09-21 09:51:32, 2020-09-21 09:52:23, Programming
2020-09-21 10:20:18, 2020-09-21 10:21:05, Writing
```

Figure 3.7: Examples of labeled time windows, collected and categorized with ActivityWatch

3.3 Analysis

We begin our analysis by labeling our raw EEG data to yield our epochs. The epochs are then transformed into 5 s windows. We then perform data cleaning, by applying a bandpass-filter to our signal and rejecting windows unsuitable for analysis. The windows are then split into training and test set splits/folds, using hold-out or LORO[3] cross-validation. The splits/folds are then used to train our classifiers, and finally we evaluate the classifier performance on the folds.

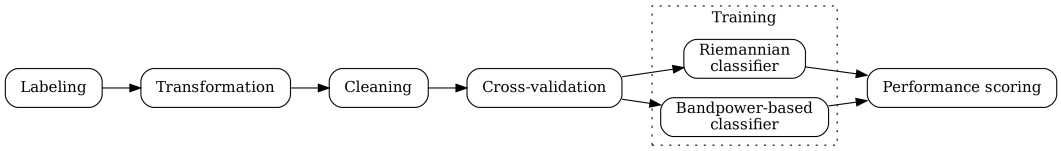


Figure 3.8: Overview of data processing pipeline used in analysis.

To perform our analysis, we have used common open source Python libraries for data analysis, like numpy [44], pandas [45], and scikit-learn [46]. In addition, we used libraries tailored specifically for working with EEG data, such as MNE [47], pyriemann [48], and YASA [49].

Source code and code notebooks for our analysis is available at: github.com/ErikBjare/thesis

3.3.1 Labelling

When labeling our data we define two levels of granularity: epochs and windows.

An epoch is a single trial, such as a single stimulus image, which can have variable length due to the experiment setup where the subject decides when to move on to the next stimulus. We label epochs using markers from our experiments.

A window is a fixed-duration slice of an epoch, such that each window matrix has the same dimensions. This enables us to more easily compute various features which require matrices to be of the same dimension. Windows inherit the label of the epoch from which they belong.

For the naturalistic device use experiment, we split the EEG data into epochs using the categories assigned by our ActivityWatch script.

For the controlled experiment, we split the EEG data into epochs using the trial markers, resulting in one epoch per stimulus.

[3] LORO

3.3.2 Data transformation

In order to classify the epochs, which have variable length due to each subject taking a different amount of time to answer each trial, we split each epoch into a fixed-duration window. After some experimentation, we found 5 s to be a suitable window size. Once we have our fixed-duration windows, we can use them to train our model and classify new samples.

Dimensions of each epoch matrix:

$$(n_{\text{samples}}, n_{\text{channels}})$$

Where n_{samples} is the total number of samples for the epoch (variable-length), and n_{channels} is the number of channels (4 for the Muse S).

Since the matrix has variable dimensions for each epoch, we split it into 5 s windows, which at the 256Hz sampling frequency of the Muse gives us 1280 samples per window. The dimensions of each window matrix is then $(1280, n_{\text{channels}})$.

The matrix of all windows thus has the following dimensions:

$$(n_{\text{windows}}, 1280, n_{\text{channels}})$$

To aggregate these window-level predictions back into the epoch-level, we will take the mean of the prediction probabilities for each window in the epoch.

We also experimented with different windowing methods to augment the data, but we ultimately decided against using them, as the methods did not seem to contribute significantly to the performance of trained classifiers.

3.3.3 Data cleaning

We reject samples that either:

1. Do not have an assigned class
2. Have a bad signal quality (as indicated by a signal variance > 40)
3. Are too short due to missing samples (such that a 5 s window cannot be constructed)

The exact number of samples rejected by each cleaning step, as well as the exact parameters used, can be viewed in the code notebooks published alongside this report.

We also perform bandpass filtering between 3 Hz and 40 Hz to eliminate powerline noise. We can see the results of the bandpass filtering in Figure 2.4 and 2.5.

3.3.4 Classification pipelines

In this section we describe our classification pipelines which use two different approaches:

1. Use bandpower features to train a common classifier algorithm (such as logistic regression, support vector machines, or random forests).
2. Use Riemannian methods to work with distances between covariance matrices by using the Riemannian metric, and then apply a common classifier algorithm.

3.3.4.1 Bandpower-based

Bandpower features are simple and commonly used in EEG research for many tasks, including the paper by Fucci et al. we seek to improve upon [4].

As a benchmark reference, we implemented classifiers which solely used bandpower features as input, to gain information of how much any improvement from classifier performance is likely due to better EEG equipment versus how much is due to from improved analysis methods.

To compute this feature, we utilized the bandpower function provided by YASA [49]. The implementation estimates the power spectral density using Welch's method for each channel, and bins them by their associated frequency band (seen in Table 2.2 on page 24).

To further enrich our feature vector, we can use ratios between two frequency bands. As this is what Fucci et al. did, we do the same.

Finally, the pipeline trains 3 models using either logistic regression, support vector machines, or random forests.

3.3.4.2 Riemannian geometry

The state of the art in many EEG classification tasks involves the use of Riemannian geometry (described in Section 2.2.1). For this, we used the open source `pyriemann` library by Alexandre Barachant[4].

Our final pipeline is defined by:

```
from sklearn.pipeline import make_pipeline
from sklearn.linear_model import LogisticRegression
from pyriemann.estimation import Covariances
```

[4] First author of the original paper to apply Riemannian geometry to EEG [50]

```
from pyriemann.spatialfilters import CSP
from pyriemann.tangentspace import TangentSpace

clf = make_pipeline(
    Covariances(),
    CSP(4, log=False),
    TangentSpace(),
    LogisticRegression(),
)
```

We do not perform any hyperparameter optimization.

3.3.5 Performance scoring

Previous studies have used *balanced accuracy* (BAC) scoring to evaluate model performance. To be able to compare results to previous studies, we do the same.

BAC deals with imbalanced datasets by evaluating the performance of the classifier on each class and combining them into a single number. It is defined as the average of recall obtained on each class, or equivalently for a binary classifier, the average of the classifier sensitivity and specificity.

For a binary classifier, a balanced accuracy score of 0.5 is no better than chance, and a score of 1 is perfect.

3.3.6 Cross Validation

We use *Leave-One-Run-Out* (LORO) cross-validation, a variation of LOGO, in order to ensure the samples used in validation are from subjects that are unseen in training. LORO validation generates n folds (one for each subject), with $n - 1$ subjects used for training in each fold, and 1 subject for validation (seen in Table 3.4).

	Subject 1	Subject 2	Subject 3	Subject 4
Fold 1				
Fold 2				
Fold 3				
Fold 4				

Table 3.4: Example of Leave-One-Run-Out cross validation with 4 subjects. For each fold, subjects marked **blue** are used for training and subjects marked **orange** are used for testing.

To aggregate performance metrics from each fold, we take the median of the performance metrics from each fold (as presented in Section 4).

Since we have one run per subject, this LORO cross-validation is effectively an out-of-subject validation, estimating the ability of the classifier to generalize across subjects.

Chapter 4

Results

In this chapter we present the results from our two different experiments.

For our code vs prose experiment we present and compare our main classifier, based on Riemannian geometry (described in Section 3.3.4.2 on page 40), versus our reference classifier, which uses the bandpower-features approach (described in Section 3.3.4.1 on page 40, similar to that used by Fucci et al). We also compare our results to the original fMRI study by Floyd et al. [3] and the EEG replication study by Fucci et al. [4].

For our naturalistic device use experiment we present performance scores for six classifiers using our Riemannian classifier (one for each pairing of our chosen class labels) to see if results from the code vs prose experiment generalize to other types of device activity.

4.1 Code vs prose task (controlled)

Table 4.1 and 4.2 show the performance we achieved for the code vs prose task, for two different subject selections. Figure 4.1 shows a detailed overview of the data, and classification results for one example subject-fold. Finally, we compare our results to previous studies in Table 4.3.

Our top-performing classifier, using Riemannian methods and cross-validated using LORO, yields a median BAC of **0.749** for window-level classification and **0.9** for epoch-level classification (seen in Table 4.2).

The **Bandpower** columns show the results for each subject-fold using the bandpower benchmark for window-level data and epoch-level data, respectively. Correspondingly, the **Riemannian** columns show the results using Riemannian geometry. The classifier using Riemann geometry tends to outperform the baseline in each fold.

We perform window-level classification by training on 5 s windows. We then achieve epoch-level classification by taking the mean of the prediction probabilities from the windows in each epoch (as described in Section 3.3.2).

	Riemannian		**Bandpower**	
Subject	Window-level	Epoch-level	Window-level	Epoch-level
#0	0.608	0.603	0.502	0.551
#1	0.802	0.864	0.666	0.677
#5	0.589	0.534	0.602	0.667
#6	0.701	0.767	0.702	0.675
#7	0.694	0.800	0.725	0.733
#8	0.547	0.542	0.546	0.625
#9	0.484	0.500	0.418	0.500
#10	0.474	0.500	0.452	0.500
Median	0.5985	0.573	0.574	0.646

Table 4.1: The balanced accuracy for each LORO subject-fold. Excluding subjects #3 and #4 due to poor signal quality. Subject #2 does not exist due to an off-by-one mistake during experiment setup (what should have been subject #2 became #3).

In Table 4.1 we see that performance is bad (no better than chance, i.e. $\boldsymbol{BAC \approx 0.5}$ as described in Section 3.3.5 on page 41) for several subjects. We investigate these and find issues with the quality and amount of data (seen in Figure 4.1 on page 46). Due to this we remove them from our dataset, and get the improved results seen in Table 4.2 on the next page. We discuss our subject selection further in Chapter 5 on page 49.

Compared to previous studies, we achieve a moderate improvement over the EEG-only classifier trained in Fucci et al., and achieve a similar performance to the fMRI study by Floyd et al. (seen in Table 4.3).

	Riemannian		Bandpower	
Subject	Window-level	Epoch-level	Window-level	Epoch-level
#0	0.673	0.727	0.511	0.541
#1	0.895	0.955	0.689	0.809
#5	0.616	0.542	0.628	0.750
#6	0.864	0.908	0.739	0.737
#7	0.749	0.900	0.733	0.733
Median	0.749	0.900	0.689	0.737

Table 4.2: The balanced accuracy for each LORO fold/subject. Excluding subjects 3, 4, 8, 9, and 10.

It should be noted that the epoch-level numbers are highly variable due to the small number of subjects (for the median) and total trials (for each subject). Therefore, we only present our best window-level results in Table 4.3.

	This study		Fucci et al.	Floyd et al.
	Riemannian	Bandpower		
Overall	0.75	0.69	0.66	0.79

Table 4.3: Result comparison between the previous studies and this study. Best balanced accuracy scores are reported. For this study, we used the best window-level score. For Fucci et al. we chose the best EEG-only score.

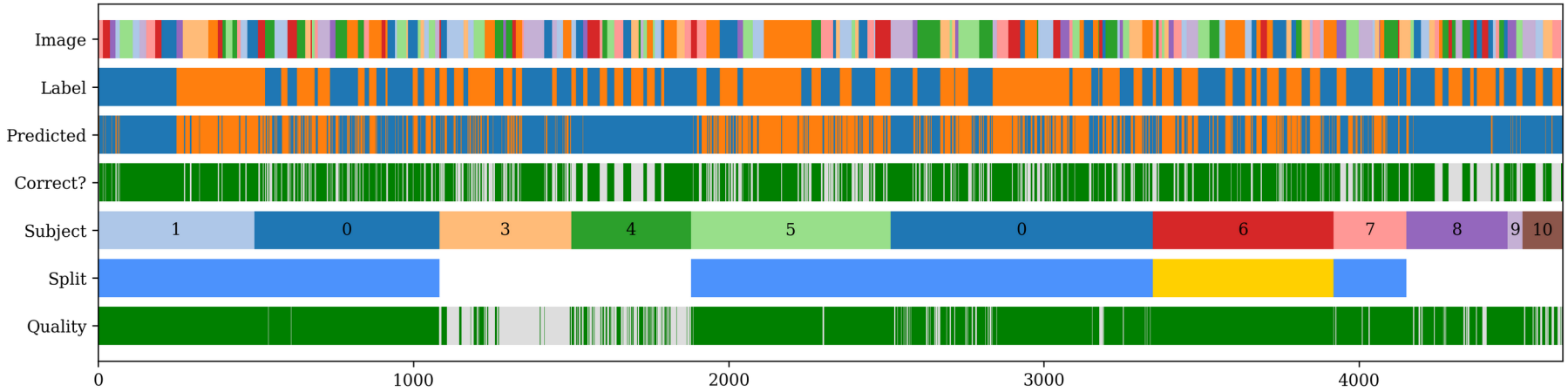


Figure 4.1: Visualization of the labeled data with classifications from one example subject-fold. Shows the *Image* (stimuli), the class *Label* for that stimuli (**blue** is code, **orange** is prose), the *Predicted* class, whether the prediction is *Correct*, the *Subject*, the *Split*/Fold (**blue** shows the training set, **yellow** the test set), and our threshold measure for signal *Quality* (**green** indicates acceptable quality). The x-axis is the window index, sorted by acquisition time.
It can be seen that (1) subjects #3 and #4 have bad signal quality, and have therefore been excluded from the training set.

(2) The subjects #9 and #10 have also been excluded from training due to issues during data collection. (3) For subject #1 the stimuli images were not shuffled. (4) Subject #0 appears twice, as they did two sessions (using unseen stimuli).

4.2 Naturalistic device activity

As described in Section 3.2.1, we collected approximately 5 hours of labeled EEG data using our labels described in Section 3.2.2. The data was collected on several different days, with a breakdown of the date and class distribution shown in Figure 4.2.

Using that data, we trained our classifiers based on Riemannian geometry for each pair of label combinations. The results for each pairing are seen in Table 4.4.

Experiment	Score	Support	Hours
Programming vs Writing	0.676	(1386, 209)	2.22h
Programming vs Twitter	0.695	(1386, 949)	3.24h
Programming vs YouTube	0.672	(1386, 266)	2.29h
Twitter vs Writing	0.833	(949, 209)	1.61h
Twitter vs YouTube	0.604	(949, 266)	1.69h
YouTube vs Writing	0.889	(266, 209)	0.66h

Table 4.4: The scores for each label pairing. The *Score* is the mean balanced accuracy of the StratifiedKFold splits. The *Support* is the number of windows for each class. *Hours* is the sum of both classes' duration.

We can see that our balanced accuracy scores are roughly in the same range as the scores achieved in our controlled code vs prose experiment. We note that our naturalistic "Programming vs Writing" classifier achieves a BAC of 0.68, which is slightly worse than the 0.75 BAC from our controlled experiment. We discuss the results further in Chapter 4 on page 43.

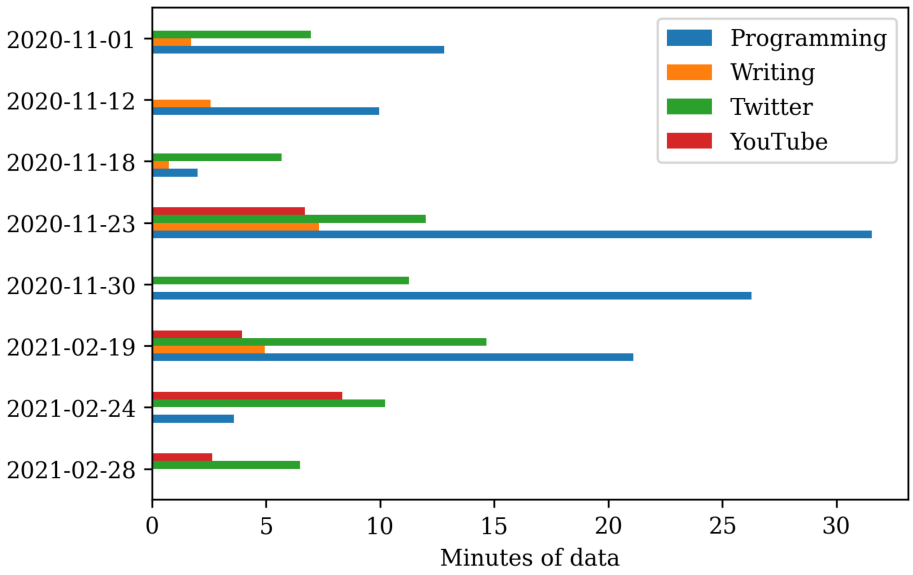


Figure 4.2: Class and date distribution of collected data.

Chapter 5

Discussion

We now move on to discussing our results and answering our two research questions. We will also discuss some threats to the validity, potential applications, ethical considerations, and a brief mention of research initiatives we have contributed to along the way.

5.1 RQ1 — Improving upon previous results in classifying code vs prose

We found that we do improve upon the best classifier performance by Fucci et al. in their EEG-only configuration [4]. However, we had to remove certain subjects from our sample, and our sample in general was smaller and more homogenous.[1] We discuss this limitation further in Section 5.3 on the following page.

Our findings also show that Riemannian methods, as used in this thesis, are better at distinguishing between the code and prose tasks than the bandpower-features approach, as used by Fucci et al. Our Riemannian score[2] was 0.75, which is better than both our bandpower-features score of 0.69, as well as the Fucci et al score of 0.66. This is in line with our expectations, as Riemannian methods are considered state-of-the-art for many EEG and BCI tasks, unlike bandpower features.

[1] The reason for this cherry-picked subject selection is due to issues with signal quality and limited data (due to spurious device disconnects).

[2] Scores taken from Table 4.3

We were not able to outperform the results by Floyd et al. [3]. This is not surprising given they were using fMRI, which has far superior spatial resolution and can therefore accurately detect elevated activity in specific brain regions over the duration of an epoch. However, our method using EEG has the benefit of high temporal resolution, allowing classification using small amounts of data, enabling near real-time applications.

5.2 RQ2 — Identify software developers' work tasks based on brain activity

We train several classifiers and achieve promising results. We were especially successful in distinguishing work (writing code) from social media (Twitter), with a BAC of 0.69.

Among the classifiers we train, we found that discerning leisure-activities from each other (such as YouTube vs Twitter, with a BAC of 0.60) is harder than discerning work from leisure (such as writing code vs Twitter). We also find that EEG is sufficient to not only pick up differences in code vs prose *comprehension* but also in *writing*, as shown by our writing code vs writing prose categories (BAC of 0.68).

We conclude that our preliminary results indicate that it is possible to discern several different device activities using EEG. However, more data from multiple subjects are needed in future work to validate the method and results. We also want to note that we only try to discern two types of activity at a time, pair by pair, and do not try to identify them among all possible work activities (a much more difficult task, left for future work).

5.3 Threats to validity

During our work we have considered several potential threats to validity. Some of these arise from our limited and biased dataset, while others are about the task and methodology.

Starting with our dataset, we collected on mostly right-handed males in their late 20s. This uniform/homogenous sample may lead to less data needed to train a classifier for that particular group, but does not necessarily generalize well to the population at large. Future studies should include a more diverse sample of participants.

We also had issues during data collection with spurious disconnects from the device, leading to data loss and incomplete experiment runs. This is a threat to the validity of the study due to not all subjects having undergone as many (or the same) trials.

Among our considerations, one threat to the validity is the stimulus images themselves (seen in Figure 3.5). In Floyd et al. the images used for prose comprehension are in fact more like a code review task, while Fucci et al. modified the images to test comprehension, instead of

ability to judge correctness. We also discovered that subjects found the prose stimulus used by Floyd et al. confusing, and it would have been preferable to use prose stimulus more like that used by Fucci et al.

The stimulus images differ on more than just content. Examples of such differences are the background color, the difference in eye saccades while reading (eyes jump around more during the code tasks, where the user may have to jump between the code and the question about it). We have not been able to discard the possibility that the front-heavy electrode placements (with reference electrode at Fpz) lead to much of the signal being from eye saccades, possibly leading the classifier to rely on differences in eye saccades instead of neural activity. Future research could evaluate the impact of eye saccades on classification performance further, either by using an eye tracker or by using an EEG device with different electrode placements.

5.4 Applications to software engineering

Gaining insight into the minds of developers at work can be used to aid and enhance the productivity of developers in several ways. As an example, it could be used to assist the developer in real time, like helping to identify developer confusion, but other applications could be imagined where a summary of the developers' brain activity during a particular commit (such as if the developer was confused, focused, tired) is included in the commit message, to help identify commits prone to introducing bugs.

Applications of our results include:

- Use the confidence in the task prediction as an alternative measurement of focus (not merely measuring that the subject was focused, but what they were focused on).
- Detecting distraction.

We also consider our novel idea of 'mental churn', a neurally integrated alternative to the measure of code churn, where not only code changes are measured, but also the *attention* a file or piece of code receives as a whole. It can be viewed as quantifying the *cognitive load* of the developer, and attributing it to the current context. This notion of mental churn could also include what response the code elicited in an engineer (confusion, focus, confidence).

5.5 Ethical considerations

When studying EEG data a range of ethical considerations arise.

- Could the data be considered personally identifiable information (PII) and thereby fall under GDPR's regulations concerning "sensitive user data"?

- How privacy sensitive are EEG recordings? Could they contain something the subject would rather keep private? (could have medical implications)
- How do we build large public datasets, while preserving participants privacy?

Companies such as Neurosity have taken an approach with their products where all the processing happens on-device, and only aggregates and classifier outputs are sent to the cloud for storage and presentation to the user. This is similar to the approach taken by ActivityWatch for device usage data, where all data is stored locally and never sent to a remote server for processing.

We believe this approach is the most privacy-preserving, but it comes at the cost of difficulty in building large datasets, as the data is no longer collected in a central repository under the control of companies or service providers. Therefore, there is a need to bridge the gap between privacy and sharing data.

Simple solutions to this problem include opt-in to data collection, which is easy to do at scale for companies offering EEG devices. But other efforts include projects such as the NeuroTech Challenge (mentioned in Section 5.6.1).

There are also more advanced solutions, such as privacy-preserving ML systems. One example of such solutions is PySyft [51], which enables private deep learning using federated learning, differential privacy, and encrypted computation (through Multi-Party Computation and Homomorphic Encryption) to work within common deep learning frameworks such as PyTorch and Tensorflow.

5.6 Democratization of neuroscience

This thesis was made possible due to the efforts of individuals and communities such as NeuroTechX to democratize neuroscience. Indeed, it is the explicit goal of the NeuroTechX eeg-notebooks [39] project to 'democratize the neuroscience experiment'. Combined with the rapid cost reduction of research-grade EEG equipment over the last decade it has enabled hobbyists to design and perform high-quality neuroscience experiments. This enables citizen scientists to contribute to data collection, an example of such an effort is the NeuroTech Challenge Series (described in Section 5.6.1).

As development of BCIs advance and the consumer market for EEG devices grow (as evidenced by new devices being released with a regular cadence by InteraXon and Neurosity) we expect to see more uses and applications of these devices.

Much of this work was made possible due to the efforts of communities such as NeuroTechX to democratize neuroscience by publishing tools for running experiments. As part of the thesis, we have contributed changes back to some of the tools used (as mentioned in Section 1.3 on page 17).

5.6.1 Crowdsourcing data

Collecting data is a significant time sink for researchers, and efforts to crowdsource data from the general public are difficult for EEG as it still requires access to the equipment and the knowledge to operate it. Crowdsourcing data also comes with new challenges. One such challenge is that the data might now be recorded by different devices, with differences in channel count, sampling rate, electrode placement, and so on. These differences can make it difficult to combine several datasets into one.

A potential solution to the problem of learning new variations from a small additional sample is to attempt cross-task or cross-subject transfer learning. In such a setup, a model is already trained on a group of subjects, or a similar task, and through a small amount of new data for a particular subject or task, the model can adapt to learn those previously unseen subjects or tasks. This is discussed further in Section 5.7.

As part of the thesis work I have contributed to the NeuroTech Challenge Series (NTCS), an effort in crowdsourcing EEG data using the experiments built in `eeg-notebooks`. The challenge can be performed by anyone with access mobile consumer EEG devices, like those from Muse, OpenBCI, and Neurosity. The project is a collaborative effort led by John Griffiths at the University of Toronto, with support from OpenBCI and NeuroTechX.

5.7 Transfer learning

An important aspect, as highlighted earlier in this thesis, is the ability of a classifier to be able to work on subjects unseen in training. Some BCI systems employ a calibration phase to achieve greater performance, but this can be time-consuming and straining for the user.

To minimize this need for calibration is a stated research goal in Khazem et al. [52], which presents what is called Riemannian Transfer Learning. They build on Riemannian geometry used in state-of-the-art classifiers, and develop a variation of MDM (explained in Section 2.2.1) called *Minimum Distance to Weighted Mean* (MDWM). The method takes a parameter λ (where $0 \leq \lambda \leq 1$) that controls how much the algorithm should rely on the class centroids learned from past subjects versus the calibration data from a new subject. This is useful in online learning contexts, where the parameter can initially start at 0 and then be incrementally adjusted towards 1. The researchers found $\lambda = 0.7$ to be a reasonable value for many tasks.

The researchers also considered using weights for each source subject as done by Kalunga et al. [53], which could be used to adjust for subject similarity. We investigated this avenue of inquiry to potentially minimize the amount of data collection needed for the model to adapt to a new subject, but in the end did not have the time to implement it.

Chapter 6

Conclusions

Our results successfully replicate previous research, showing that it is possible to distinguish between reading code and prose using EEG, and improves upon the state of the art in this regard by achieving a roughly ~75% balanced accuracy (using LORO cross-validation). However, the reader should note limitations of our study as discussed in Section 5.3 on page 50.

Furthermore, our naturalistic experiment results show that it was possible to distinguish many other device activities from each other using consumer-grade EEG devices. Among these some data seems to suggest that EEG is sufficient to not only pick up differences in code vs prose *comprehension* but also in *writing*.

We conclude by discussing how the experiment setup and analysis method could be improved in future work, as well as the need for more data and diverse subjects, to ensure that the results are robust and generalize to the population at large.

6.1 Future work

The amount of future work to do in this area is substantial, as the author has noticed during the work fighting to keep the scope of this master's thesis under control. We will begin this section by focusing on the particular code vs prose task used in the experiment, then we will discuss our naturalistic device use classification, and finally some words on the future prospects of low-cost functional brain imaging systems.

6.1.1 Improvements to code vs prose task

To improve the robustness and easy of use for the code vs prose task, we suggest that future researchers attempting the same experiment should make a few changes:

- As mentioned in the Method chapter, this study uses prose *review* images from Floyd et al., as opposed to the prose *comprehension* images (in Italian) used by Fucci et al. Ideally, one should create an English prose comprehension set of stimuli images, similar to the ones used by Fucci, as they would better represent the intended prose comprehension task.
- Try to use even better EEG equipment. It would be interesting to investigate if higher density EEG setups could improve classification performance further.

In addition to changes for the experiment itself, the analysis of the data from the code vs prose task could also be integrated the into the BCI benchmark-suite `moabb`[1] to make it easier to replicate the results and evaluate new analysis methods. As mentioned, the study would also benefit from more data collection, possibly through the NeuroTech Challenge Series (mentioned in Section 5.6.1).

6.1.2 Improvements to naturalistic device use task

For our naturalistic device use classification task, we have considered a few possible improvements that we, in the interest of time and managing scope, left out for future work.

- Collect data from multiple subjects, and use more broadly defined activities to better suit different subjects' device usage.
- Make use of the input watcher we developed (mentioned in Section 3.2.2 on page 37), to separate code and prose reading/comprehension from active writing.
- Combining with classifiers of emotion and other brain states, to understand associations between them and different device activities.

We are curious about combining device use data with EEG classifiers of emotion[2] or other brain states, to try to monitor how we feel during different activities. This could potentially be introduced as another feature for time tracking software like ActivityWatch, which could then not only be used to see how you spent your time, but also how you felt during that time.

It would also be of interest to implement more tasks engaging specific cognitive processes in `eeg-notebooks`. As an example, implementing working memory-intensive tasks such as n-back, or simple tasks involving basic arithmetic operations.

[1] Mother of All BCI Benchmarks, https://github.com/NeuroTechX/moabb

[2] Notably difficult, in part due to the subjective nature, and difficulty to consistently elicit in subjects

We encourage researchers to experiment with our naturalistic device use classification method, as there is a lot of new ground to cover in that area.

6.1.3 Integrating with software engineering research

The work on using EEG and other forms of quantified psychophysiology in software engineering is a budding and growing field of inquiry, but methods and equipment are not yet mature for widespread use. We believe researchers would do well to publish more applications of their work, such as open source tools for monitoring the developers' psychophysiology (as done in part by this study) to invite more software engineering practitioners into the field.

In conclusion, we see a growing abundance of research opportunities in the intersection of neuroscience and software engineering. We expect the intersection of the two fields to continue to expand, and over time commercialize further to enable widespread access to reliable EEG devices and software that can support software engineers in their work.

Bibliography

[1] Phattarapong Sawangjai et al. “Consumer Grade EEG Measuring Sensors as Research Tools: A Review”. In: *IEEE Sensors Journal* 20.8 (Apr. 2020), pp. 3996–4024. ISSN: 1558-1748. DOI: 10.1109/JSEN.2019.2962874.

[2] *ActivityWatch: Open Source Automated Time Tracker*. In collab. with Erik Bjäreholt and Johan Bjäreholt. ActivityWatch, May 23, 2020. DOI: 10.5281/zenodo.3794886. URL: https://github.com/ActivityWatch/activitywatch (visited on 05/23/2020).

[3] Benjamin Floyd, Tyler Santander, and Westley Weimer. “Decoding the Representation of Code in the Brain: An fMRI Study of Code Review and Expertise”. In: *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*. 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE). May 2017, pp. 175–186. DOI: 10.1109/ICSE.2017.24.

[4] Davide Fucci et al. “A Replication Study on Code Comprehension and Expertise Using Lightweight Biometric Sensors”. In: *2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC)*. 2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC). May 2019, pp. 311–322. DOI: 10.1109/ICPC.2019.00050.

[5] Thomas Fritz et al. “Using Psycho-Physiological Measures to Assess Task Difficulty in Software Development”. In: *Proceedings of the 36th International Conference on Software Engineering*. ICSE 2014. New York, NY, USA: Association for Computing Machinery, May 31, 2014, pp. 402–413. ISBN: 978-1-4503-2756-5. DOI: 10.1145/2568225.2568266. URL: https://doi.org/10.1145/2568225.2568266 (visited on 09/25/2021).

[6] Manuela Züger et al. “Reducing Interruptions at Work: A Large-Scale Field Study of FlowLight”. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI ’17. Denver, Colorado, USA: Association for Computing Machinery, May 2, 2017, pp. 61–72. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025662. URL: https://doi.org/10.1145/3025453.3025662 (visited on 06/13/2020).

[7] Daniela Girardi et al. *Recognizing Developers' Emotions While Programming*. Jan. 24, 2020. DOI: 10.1145/3377811.3380374. arXiv: 2001.09177 [cs]. URL: http://arxiv.org/abs/2001.09177 (visited on 09/23/2020).

[8] Melanie Swan. "The Quantified Self: Fundamental Disruption in Big Data Science and Biological Discovery". In: *Big Data* 1.2 (June 1, 2013), pp. 85–99. ISSN: 2167-6461. DOI: 10.1089/big.2012.0002. URL: https://www.liebertpub.com/doi/full/10.1089/big.2012.0002 (visited on 11/16/2021).

[9] Gin S. Malhi et al. "The Promise of Digital Mood Tracking Technologies: Are We Heading on the Right Track?" In: *Evidence-Based Mental Health* 20.4 (Nov. 1, 2017), pp. 102–107. ISSN: 1362-0347, 1468-960X. DOI: 10.1136/eb-2017-102757. pmid: 28855245. URL: https://ebmh.bmj.com/content/20/4/102 (visited on 09/02/2021).

[10] Keum-Shik Hong, Noman Naseer, and Yun-Hee Kim. "Classification of Prefrontal and Motor Cortex Signals for Three-Class fNIRS–BCI". In: *Neuroscience Letters* 587 (Feb. 5, 2015), pp. 87–92. ISSN: 0304-3940. DOI: 10.1016/j.neulet.2014.12.029. URL: http://www.sciencedirect.com/science/article/pii/S0304394014009744 (visited on 05/22/2020).

[11] Gary H. Glover. "Overview of Functional Magnetic Resonance Imaging". In: *Neurosurgery clinics of North America* 22.2 (Apr. 2011), pp. 133–139. ISSN: 1042-3680. DOI: 10.1016/j.nec.2010.11.001. pmid: 21435566. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3073717/ (visited on 09/29/2021).

[12] D. J. McFarland and J. R. Wolpaw. "EEG-Based Brain–Computer Interfaces". In: *Current Opinion in Biomedical Engineering*. Synthetic Biology and Biomedical Engineering / Neural Engineering 4 (Dec. 1, 2017), pp. 194–200. ISSN: 2468-4511. DOI: 10.1016/j.cobme.2017.11.004. URL: http://www.sciencedirect.com/science/article/pii/S246845111730082X (visited on 05/13/2020).

[13] Mansoureh Fahimi Hnazaee et al. "Localization of Deep Brain Activity with Scalp and Subdural EEG". In: *NeuroImage* 223 (Dec. 1, 2020), p. 117344. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2020.117344. URL: https://www.sciencedirect.com/science/article/pii/S1053811920308302 (visited on 05/25/2021).

[14] Martin Spüler. "A High-Speed Brain-Computer Interface (BCI) Using Dry EEG Electrodes". In: *PLoS ONE* 12.2 (Feb. 22, 2017). ISSN: 1932-6203. DOI: 10.1371/journal.pone.0172400. pmid: 28225794. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5321409/ (visited on 05/13/2020).

[15] Bendong Zhao et al. "Convolutional Neural Networks for Time Series Classification". In: *Journal of Systems Engineering and Electronics* 28.1 (Feb. 2017), pp. 162–169. ISSN: 1004-4132. DOI: 10.21629/JSEE.2017.01.18.

[16] Robin Tibor Schirrmeister et al. "Deep Learning with Convolutional Neural Networks for EEG Decoding and Visualization". In: *Human Brain Mapping* 38.11 (2017), pp. 5391–5420. ISSN: 1097-0193. DOI: 10.1002/hbm.23730. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.23730 (visited on 05/22/2020).

[17] Jinpeng Li, Zhaoxiang Zhang, and Huiguang He. "Hierarchical Convolutional Neural Networks for EEG-Based Emotion Recognition". In: *Cognitive Computation* 10.2 (Apr. 1, 2018), pp. 368–380. ISSN: 1866-9964. DOI: 10.1007/s12559-017-9533-x. URL: https://doi.org/10.1007/s12559-017-9533-x (visited on 05/22/2020).

[18] *OpenBCI: An Open Source Brain-Computer Interface For Makers*. Kickstarter. URL: https://www.kickstarter.com/projects/openbci/openbci-an-open-source-brain-computer-interface-fo (visited on 07/07/2021).

[19] *FreeEEG32*. Crowd Supply. URL: https://www.crowdsupply.com/neuroidss/freeeeg32 (visited on 07/07/2021).

[20] Chantel S. Prat et al. "Relating Natural Language Aptitude to Individual Differences in Learning Programming Languages". In: *Scientific Reports* 10.1 (1 Mar. 2, 2020), p. 3817. ISSN: 2045-2322. DOI: 10.1038/s41598-020-60661-8. URL: https://www.nature.com/articles/s41598-020-60661-8 (visited on 05/25/2021).

[21] Anna A Ivanova et al. "Comprehension of Computer Code Relies Primarily on Domain-General Executive Brain Regions". In: *eLife* 9 (Dec. 15, 2020). Ed. by Andrea E Martin et al., e58906. ISSN: 2050-084X. DOI: 10.7554/eLife.58906. URL: https://doi.org/10.7554/eLife.58906 (visited on 12/16/2020).

[22] *Neurological Divide: An fMRI Study of Prose and Code Writing (ICSE 2020 - Technical Papers) - ICSE 2020*. URL: https://conf.researchr.org/details/icse-2020/icse-2020-papers/7/Neurological-Divide-An-fMRI-Study-of-Prose-and-Code-Writing (visited on 02/11/2021).

[23] Maarten H. W. Selfhout et al. "Different Types of Internet Use, Depression, and Social Anxiety: The Role of Perceived Friendship Quality". In: *Journal of Adolescence* 32.4 (Aug. 1, 2009), pp. 819–833. ISSN: 0140-1971. DOI: 10.1016/j.adolescence.2008.10.011. URL: http://www.sciencedirect.com/science/article/pii/S0140197108001218 (visited on 05/13/2020).

[24] Dhavan Shah et al. "Nonrecursive Models of Internet Use and Community Engagement: Questioning Whether Time Spent Online Erodes Social Capital". In: *Journalism & Mass Communication Quarterly* 79.4 (Dec. 1, 2002), pp. 964–987. ISSN: 1077-6990. DOI: 10.1177/107769900207900412. URL: https://doi.org/10.1177/107769900207900412 (visited on 05/13/2020).

[25] Gloria Mark, Yiran Wang, and Melissa Niiya. "Stress and Multitasking in Everyday College Life: An Empirical Study of Online Activity". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: Association for Computing Machinery, Apr. 26, 2014, pp. 41–50. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557361. URL: https://doi.org/10.1145/2556288.2557361 (visited on 05/22/2020).

[26] Chiungjung Huang. "Time Spent on Social Network Sites and Psychological Well-Being: A Meta-Analysis". In: *Cyberpsychology, Behavior, and Social Networking* 20.6 (June 1, 2017), pp. 346–354. ISSN: 2152-2715. DOI: 10.1089/cyber.2016.0758. URL: https://www.liebertpub.com/doi/10.1089/cyber.2016.0758 (visited on 05/13/2020).

[27] Kaveri Subrahmanyam et al. “The Impact of Computer Use on Children’s and Adolescents’ Development”. In: *Journal of Applied Developmental Psychology*. Children in the Digital Age 22.1 (Jan. 1, 2001), pp. 7–30. ISSN: 0193-3973. DOI: 10.1016/S0193-3973(00)00063-0. URL: http://www.sciencedirect.com/science/article/pii/S0193397300000630 (visited on 05/13/2020).

[28] Young-Ho Kim et al. “TimeAware: Leveraging Framing Effects to Enhance Personal Productivity”. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI ’16. San Jose, California, USA: Association for Computing Machinery, May 7, 2016, pp. 272–283. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858428. URL: https://doi.org/10.1145/2858036.2858428 (visited on 05/13/2020).

[29] Dorian Peters, Rafael A. Calvo, and Richard M. Ryan. “Designing for Motivation, Engagement and Wellbeing in Digital Experience”. In: *Frontiers in Psychology* 9 (2018). ISSN: 1664-1078. DOI: 10.3389/fpsyg.2018.00797. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2018.00797/full?report=reader (visited on 05/23/2020).

[30] *RescueTime: Automatic Time-Tracking Software*. URL: https://www.rescuetime.com/ (visited on 05/23/2020).

[31] *Hubstaff: Time Tracking and Productivity Monitoring Tool*. URL: https://hubstaff.com/ (visited on 05/23/2020).

[32] John Rooksby et al. “Personal Tracking of Screen Time on Digital Devices”. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI ’16. San Jose, California, USA: Association for Computing Machinery, May 7, 2016, pp. 284–296. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858055. URL: https://doi.org/10.1145/2858036.2858055 (visited on 05/13/2020).

[33] Roger Vieira and Kleinner Farias. *On the Usage of Psychophysiological Data in Software Engineering: An Extended Systematic Mapping Study*. May 28, 2021. arXiv: 2105.14059 [cs]. URL: http://arxiv.org/abs/2105.14059 (visited on 06/07/2021).

[34] F. Lotte et al. “A Review of Classification Algorithms for EEG-Based Brain–Computer Interfaces: A 10 Year Update”. In: *Journal of Neural Engineering* 15.3 (Apr. 2018), p. 031005. ISSN: 1741-2552. DOI: 10.1088/1741-2552/aab2f2. URL: https://doi.org/10.1088/1741-2552/aab2f2 (visited on 07/07/2021).

[35] Alexandre Barachant et al. “Common Spatial Pattern Revisited by Riemannian Geometry”. In: *2010 IEEE International Workshop on Multimedia Signal Processing*. 2010 IEEE 12th International Workshop on Multimedia Signal Processing (MMSP). Saint-Malo, France: IEEE, Oct. 2010, pp. 472–476. ISBN: 978-1-4244-8110-1. DOI: 10.1109/MMSP.2010.5662067. URL: http://ieeexplore.ieee.org/document/5662067/ (visited on 10/06/2021).

[36] Wolfgang Förstner and Boudewijn Moonen. “A Metric for Covariance Matrices”. In: *Geodesy-The Challenge of the 3rd Millennium*. Ed. by Erik W. Grafarend, Friedrich W. Krumm, and Volker S. Schwarze. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 299–309. ISBN: 978-3-642-07733-3 978-3-662-05296-9. DOI: 10.1007/978-3-662-05296-9_31. URL: http://link.springer.com/10.1007/978-3-662-05296-9_31 (visited on 07/07/2021).

[37] Marco Congedo, Alexandre Barachant, and Rajendra Bhatia. "Riemannian Geometry for EEG-Based Brain-Computer Interfaces; a Primer and a Review". In: *Brain-Computer Interfaces* 4.3 (July 3, 2017), pp. 155–174. ISSN: 2326-263X. DOI: 10.1080/2326263X.2017.1297192. URL: https://doi.org/10.1080/2326263X.2017.1297192 (visited on 10/11/2021).

[38] Olave E. Krigolson et al. "Choosing MUSE: Validation of a Low-Cost, Portable EEG System for ERP Research". In: *Frontiers in Neuroscience* 11 (Mar. 10, 2017). ISSN: 1662-4548. DOI: 10.3389/fnins.2017.00109. pmid: 28344546. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5344886/ (visited on 04/02/2021).

[39] Alexandre Barachant and John Griffiths. *Eeg-Notebooks*. NeuroTechX, Sept. 20, 2020. URL: https://github.com/NeuroTechX/eeg-notebooks (visited on 09/21/2020).

[40] Alexandre Barachant et al. *muse-lsl*. May 2019. DOI: 10.5281/zenodo.3228861. URL: https://doi.org/10.5281/zenodo.3228861.

[41] *Brainflow*. brainflow-dev, Sept. 13, 2020. URL: https://github.com/brainflow-dev/brainflow (visited on 09/13/2020).

[42] Jonathan Peirce et al. "PsychoPy2: Experiments in Behavior Made Easy". In: *Behavior Research Methods* 51.1 (Feb. 1, 2019), pp. 195–203. ISSN: 1554-3528. DOI: 10.3758/s13428-018-01193-y. URL: https://doi.org/10.3758/s13428-018-01193-y (visited on 12/16/2020).

[43] Andrej Karpathy. *Ulogme*. May 8, 2016. URL: https://github.com/karpathy/ulogme (visited on 07/07/2021).

[44] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[45] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

[46] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[47] *Mne-Python*. MNE tools for MEG and EEG data analysis, Sept. 12, 2020. URL: https://github.com/mne-tools/mne-python (visited on 09/13/2020).

[48] Alexandre Barachant et al. *alexandrebarachant/pyRiemann*. Version v0.2.6. Mar. 2020. DOI: 10.5281/zenodo.3715511. URL: https://doi.org/10.5281/zenodo.3715511.

[49] Raphael Vallat and Nikola Jajcay. *Yasa*. Zenodo, Nov. 4, 2020. DOI: 10.5281/zenodo.4244889. URL: https://zenodo.org/record/4244889 (visited on 12/16/2020).

[50] Alexandre Barachant et al. "Classification of Covariance Matrices Using a Riemannian-Based Kernel for BCI Applications". In: *Neurocomputing* 112 (July 1, 2013), pp. 172–178. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2012.12.039. URL: https://doi.org/10.1016/j.neucom.2012.12.039 (visited on 12/01/2020).

[51] Andrew Trask. *PySyft - Code for Computing on Data You Do Not Own and Cannot See*. OpenMined, Dec. 3, 2021. URL: https://github.com/OpenMined/PySyft (visited on 12/03/2021).

[52] Salim Khazem et al. "Minimizing Subject-Dependent Calibration for BCI with Riemannian Transfer Learning". In: *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*. 2021 10th International IEEE/EMBS Conference on Neural Engineering (NER). May 2021, pp. 523–526. DOI: 10.1109/NER49283.2021.9441279.

[53] Emmanuel K. Kalunga, Sylvain Chevallier, and Quentin Barthélemy. "Transfer Learning for SSVEP-Based BCI Using Riemannian Similarities Between Users". In: *2018 26th European Signal Processing Conference (EUSIPCO)*. 2018 26th European Signal Processing Conference (EUSIPCO). Sept. 2018, pp. 1685–1689. DOI: 10.23919/EUSIPCO.2018.8553441.

Further Reading

Sebastian Baltes and Paul Ralph. *Sampling in Software Engineering Research: A Critical Review and Guidelines*. Oct. 20, 2021. arXiv: 2002.07764 [cs]. URL: http://arxiv.org/abs/2002.07764 (visited on 11/30/2021).

Hubert Banville et al. "Uncovering the Structure of Clinical EEG Signals with Self-Supervised Learning". In: *Journal of Neural Engineering* (2020). ISSN: 1741-2552. DOI: 10.1088/1741-2552/abca18. URL: http://iopscience.iop.org/article/10.1088/1741-2552/abca18 (visited on 12/02/2020).

Alexandre Barachant et al. "Multiclass Brain–Computer Interface Classification by Riemannian Geometry". In: *IEEE Transactions on Biomedical Engineering* 59.4 (Apr. 2012), pp. 920–928. ISSN: 1558-2531. DOI: 10.1109/TBME.2011.2172210.

Tsung-Sheng Chang and Wei-Hung Hsiao. "Time Spent on Social Networking Sites: Understanding User Behavior and Social Capital". In: *Systems Research and Behavioral Science* 31.1 (2014), pp. 102–114. ISSN: 1099-1743. DOI: 10.1002/sres.2169. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/sres.2169 (visited on 05/13/2020).

Alexander Craik, Yongtian He, and Jose L. Contreras-Vidal. "Deep Learning for Electroencephalogram (EEG) Classification Tasks: A Review". In: *Journal of Neural Engineering* 16.3 (Apr. 2019), p. 031001. ISSN: 1741-2552. DOI: 10.1088/1741-2552/ab0ab5. URL: https://doi.org/10.1088%2F1741-2552%2Fab0ab5 (visited on 10/29/2020).

Gianluca Di Flumeri et al. "The Dry Revolution: Evaluation of Three Different EEG Dry Electrode Types in Terms of Signal Spectral Features, Mental States Classification and Usability". In: *Sensors (Basel, Switzerland)* 19.6 (Mar. 19, 2019). ISSN: 1424-8220. DOI: 10.3390/s19061365. pmid: 30893791. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6470960/ (visited on 05/13/2020).

Hermann Hinrichs et al. “Comparison between a Wireless Dry Electrode EEG System with a Conventional Wired Wet Electrode EEG System for Clinical Applications”. In: *Scientific Reports* 10.1 (Mar. 23, 2020), pp. 1–14. ISSN: 2045-2322. DOI: 10.1038/s41598-020-62154-0. URL: https://www.nature.com/articles/s41598-020-62154-0 (visited on 05/13/2020).

Ji Young Jung et al. “Caffeine Maintains Arousal Level and Prevents Change of Electroencephalogram Spectral Powers with Time at Rest”. In: *Journal of Korean Sleep Research Society* 11.1 (June 30, 2014), pp. 5–10. ISSN: 1738-608X, 2288-4912. DOI: 10.13078/jksrs.14002. URL: http://www.e-jsm.org/journal/view.php?number=171 (visited on 08/03/2020).

G. Gopan K, N. Sinha, and D. B. Jayagopi. “Alcoholic EEG Analysis Using Riemann Geometry Based Framework”. In: *2019 27th European Signal Processing Conference (EUSIPCO)*. 2019 27th European Signal Processing Conference (EUSIPCO). Sept. 2019, pp. 1–5. DOI: 10.23919/EUSIPCO.2019.8902506.

Ji Peng Koh. “Brain Computer Interface via EEG Signals”. In: (2018). URL: https://dr.ntu.edu.sg//handle/10356/73955 (visited on 05/13/2020).

Arnd Meiser et al. “The Sensitivity of Ear-EEG: Evaluating the Source-Sensor Relationship Using Forward Modeling”. In: *Brain Topography* (Aug. 24, 2020). ISSN: 1573-6792. DOI: 10.1007/s10548-020-00793-2. URL: https://doi.org/10.1007/s10548-020-00793-2 (visited on 10/05/2020).

Andrew Melnik et al. “Systems, Subjects, Sessions: To What Extent Do These Factors Influence EEG Data?” In: *Frontiers in Human Neuroscience* 11 (2017). ISSN: 1662-5161. DOI: 10.3389/fnhum.2017.00150. URL: https://www.frontiersin.org/articles/10.3389/fnhum.2017.00150/full (visited on 09/21/2020).

Christian Mühl et al. “A Survey of Affective Brain Computer Interfaces: Principles, State-of-the-Art, and Challenges”. In: *Brain-Computer Interfaces* 1.2 (Apr. 3, 2014), pp. 66–84. ISSN: 2326-263X. DOI: 10.1080/2326263X.2014.912881. URL: https://doi.org/10.1080/2326263X.2014.912881 (visited on 05/22/2020).

Bandpower of an EEG Signal - Raphaelvallat.Com/. URL: https://raphaelvallat.com/bandpower.html (visited on 04/26/2021).

Braindecode. braindecode, Apr. 19, 2021. URL: https://github.com/braindecode/braindecode (visited on 04/22/2021).

Search: EEG | Kaggle. URL: https://www.kaggle.com/search?q=eeg (visited on 09/03/2020).

Timeflux. URL: https://timeflux.io/ (visited on 09/21/2020).

PeterPutman. *The Effects of a Single Administration of a Moderate Dose of Caffeine on Cognitive Control and Spontaneous EEG Theta/Beta Ratio*. Clinical trial registration NCT02940808. clinicaltrials.gov, Mar. 7, 2017. URL: https://clinicaltrials.gov/ct2/show/NCT02940808 (visited on 07/31/2020).

Francesca Pizzo et al. “Deep Brain Activities Can Be Detected with Magnetoencephalography”. In: *Nature Communications* 10.1 (1 Feb. 27, 2019), p. 971. ISSN: 2041-1723. DOI: 10.1038/s41467-019-08665-5. URL: https://www.nature.com/articles/s41467-019-08665-5 (visited on 05/25/2021).

Stevche Radevski, Hideaki Hata, and Kenichi Matsumoto. "Real-Time Monitoring of Neural State in Assessing and Improving Software Developers' Productivity". In: *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering*. 2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE). Florence, Italy: IEEE, May 2015, pp. 93–96. ISBN: 978-1-4673-7031-8. DOI: 10.1109/CHASE.2015.28. URL: http://ieeexplore.ieee.org/document/7166096/ (visited on 09/25/2021).

John Rooksby et al. "Personal Tracking as Lived Informatics". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: Association for Computing Machinery, Apr. 26, 2014, pp. 1163–1172. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557039. URL: https://doi.org/10.1145/2556288.2557039 (visited on 05/13/2020).

Yannick Roy et al. "Deep Learning-Based Electroencephalography Analysis: A Systematic Review". In: *Journal of Neural Engineering* 16.5 (Aug. 2019), p. 051001. ISSN: 1741-2552. DOI: 10.1088/1741-2552/ab260c. URL: https://doi.org/10.1088%2F1741-2552%2Fab260c (visited on 10/29/2020).

Lamyaa Sadouk. "CNN Approaches for Time Series Classification". In: *Time Series Analysis - Data, Methods, and Applications* (Nov. 5, 2018). DOI: 10.5772/intechopen.81170. URL: https://www.intechopen.com/books/time-series-analysis-data-methods-and-applications/cnn-approaches-for-time-series-classification (visited on 05/22/2020).

Axel Uran et al. *Applying Transfer Learning To Deep Learned Models For EEG Analysis*. July 2, 2019. arXiv: 1907.01332 [cs, eess, stat]. URL: http://arxiv.org/abs/1907.01332 (visited on 11/23/2020).

Laurent Vézard et al. "EEG Classification for the Detection of Mental States". In: *Applied Soft Computing* 32 (July 1, 2015), pp. 113–131. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2015.03.028. URL: http://www.sciencedirect.com/science/article/pii/S1568494615001866 (visited on 11/10/2020).

Zitong Wan et al. "A Review on Transfer Learning in EEG Signal Analysis". In: *Neurocomputing* 421 (Jan. 15, 2021), pp. 1–14. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2020.09.017. URL: http://www.sciencedirect.com/science/article/pii/S0925231220314223 (visited on 11/23/2020).

Yu Zhang et al. "Multi-Kernel Extreme Learning Machine for EEG Classification in Brain-Computer Interfaces". In: *Expert Systems with Applications* 96 (Apr. 15, 2018), pp. 302–310. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2017.12.015. URL: http://www.sciencedirect.com/science/article/pii/S0957417417308291 (visited on 05/13/2020).

Glossary

ActivityWatch A free and open-source automated time tracker, that tracks what you do on your computer. 9, 10

Quantified Self A cultural phenomenon of self-tracking with technology to achieve "self-knowledge through numbers". Also a particular community of the same name. 10

Acronyms

BAC balanced accuracy 41, 44, 47

BCI Brain-Computer Interface 11

CNN Convolutional Neural Network 11

ECoG Electrocorticography 21

EEG Electroencephalography 5, 9, 11, 19, 21, 23, 25

ERP Event-Related Potential 11, 19, 20

fMRI Functional Magnetic Resonance Imaging 11, 43

fNIRS Functional Near-Infrared Spectroscopy 11

HCNN Hierarchical Convolutional Neural Network 11

HEG Hemoencephalography 11

iEEG Intracranial Electroencephalography 21

LOGO Leave-One-Group-Out 41

LORO Leave-One-Run-Out 38, 41, 44

MCN Modified Combinatorial Nomenclature 21

MRI Magnetic Resonance Imaging 9

VEP Visual Evoked Potential 11