Using deep learning to predict children's age and risk of dyslexia from the event related potential

Floris Pauwels

Supervisor and Examinor:
dr. Hugo Schnack

Supervisor at the eScience Center:
Candace Makeda Moore, MD

Second Examinor:
prof. dr. Frank Wijnen

Universiteit Utrecht

Master Thesis
Artificial Intelligence
21 November 2022
Student Nr. 5698634

Abstract

This thesis uses electroencephalography (EEG) data to predict the age and risk of developmental dyslexia of young children. It is useful to diagnose dyslexia at a young age to conduct interventions that reduce reading and writing difficulties later in life.

The ePodium and the Dutch Dyslexia Project (DDP) dataset are used. Both these datasets use the auditory oddball paradigm to elicit a standard and deviant Event Related Potential (ERP). The EEG data is pre-processed with the autoreject library, which removes many artifacts in the data. The cleaned trials around an auditory event are averaged to create ERPs. These ERPs are used by deep learning models to predict the age and risk of dyslexia from patterns within the data.

The results of a previous master thesis affiliated with the ePodium project are reproduced. The thesis trained a deep learning model that found a correlation between age and the standard ERP signals of children in the DDP dataset. Reproducing the results confirms that the encoder model is the state-of-the-art model on age prediction from ERPs.

Trained models can already make reasonable age predictions from a small subset of the total standard trials within an experiment. Also, adding a significant amount of Gaussian noise to each ERP signal does not significantly alter the performance of the models. These observations indicate that the models base their predictions on the global pattern of the ERP and not from local voltage differences in the millisecond range.

Transfer learning between datasets is possible as models trained on the DDP dataset found a correlation between the ERPs from the standard event and age within the ePodium dataset despite the differences between the two datasets. There was a difference in results between models that used standardised and non-standardised ERPs.

The encoder model was unable to find patterns for predicting age and dyslexia from the ePodium dataset. This dataset may be too small for deep learning to make predictions. Some solutions to this problem can be to use more data-efficient methods like time-frequency analysis on raw EEG data or to create simulated data to artificially increase the size of the dataset.

Contents

1 Introduction

Developmental dyslexia is a learning disorder that causes difficulty in reading and writing. Dyslexia is currently diagnosed after a child is expected to read and write. It is however useful to diagnose dyslexia at a younger age. Young children at risk of dyslexia can be provided extra classroom instructions in letter-sound correspondences to reduce difficulty in reading and writing later in life [1].

Developmental dyslexia is related to a speech processing deficit in infants as young as two months [2]. The brains of infants that develop dyslexia respond differently to syllable sounds than typical infants. This means that dyslexia could theoretically be predicted at a young age by measuring abnormalities in the brain response when infants are listening to syllable sounds.

Brain activity can be detected with *electroencephalography* (EEG). EEG signals contain electrical activity of the brain from electrodes that measure voltages on the head. EEG has previously been successful at finding deviations in brain responses in multiple disorders such as autism [3] and schizophrenia [4].

EEG research that uses *deep learning* for classification EEG data has grown exponentially over the last few years [5]. For example, deep learning has been applied on the classification of different sleep stages, and the detection and prediction of seizures. Deep learning is a machine learning technique that is capable of learning complex patterns from large amounts of data. The main goal of this thesis is to investigate whether deep learning can predict dyslexia from the brain signals of young children before they learn to read.

ePodium Project

This thesis is part of the project *ePODIUM: early Prediction Of Dyslexia in Infants Using Machine learning.* The project is a collaboration between researchers from Utrecht University, UMC Utrecht, and the Netherlands eScience Center (NLeSC) in Amsterdam. The goal of the ePodium project is to explore if EEG data measured in infancy can predict the occurrence of later literacy difficulties in individual children.

As a part of the ePodium project, a master student had been successful at predicting the age of children between 11 and 47 months from EEG signals [6]. This was done by applying deep learning models to the *Dutch Dyslexia Program* (DDP) dataset. The best performing model was the *encoder* model which could predict the age of children between 11 and 47 months with a mean absolute error of approximately five months.

The ePodium project has made a new dataset in which EEG data is collected from a total of 129 toddlers. In this experiment, the toddlers are measured twice between the age of 16 and 24 months with a three month gap between the two measurements [7].

The experiment is an auditory oddball task, which means that the subject listens to frequent and infrequent sounds, with a short interval between each sound. The difference in brain response resulting from the frequent and infrequent stimuli can be determined from the measurements. The hypothesis is that this response difference is less prominent for children that develop dyslexia, since auditory sensory processing is impaired in these children [8]. If the hypothesis is correct, it is possible to distinguish between dyslexic and non-dyslexic children from the EEG data of this experiment at a young age.

The dataset contains scores of dyslexia tests from the parents, but does not contain data on whether the children will actually develop dyslexia. Nevertheless, children with dyslexic parents have a higher risk of developing dyslexia themselves [9]. In the thesis the objective of the deep learning models is to predict this parental risk of dyslexia.

Thesis Objectives

The main goal of the ePodium project is to predict dyslexia from EEG data of toddlers with deep learning. To get closer to the goal, this thesis formulates intermediate steps.

The first step is to reproduce the results of the previous thesis that predicted age by applying the encoder deep learning model to the DDP dataset. Children with a family-history of dyslexia have significantly reduced gray matter in some parts of the brain [10]. This may have an effect on the age prediction of the models.

Secondly, the model performance is compared for multiple input modifications, such as the amount of artificial noise and whether the data is standardised. Comparing the performance differences as a result of these setting variations will give a better insight in optimal parameters for future research.

Thirdly, the best performing age model on the DDP dataset is applied on the ePodium dataset, to test if the model is transferable between datasets. The main advantage of transfer learning is that one model can be trained on one dataset and used on another dataset for additional training and prediction.

2 Theory

2.1 Electroencephalography (EEG)

Electroencephalography (EEG) is a technique that measures and records the electrical activity of the brain. This technique measures the electrical activity from multiple electrode sensors placed on the head. These electrodes pick up tiny electrical voltages caused by the firing of many neurons in the brain. The voltage from each electrode is amplified and subsequently recorded on a computer. A measurement can be performed in the millisecond range, to produce hundreds or even thousands of voltage measurements every second [11]. The voltage as a result of noise is often many times higher than the voltage originating from active neurons, so the EEG data usually requires multiple processing techniques in order to extract useful information on brain activity.

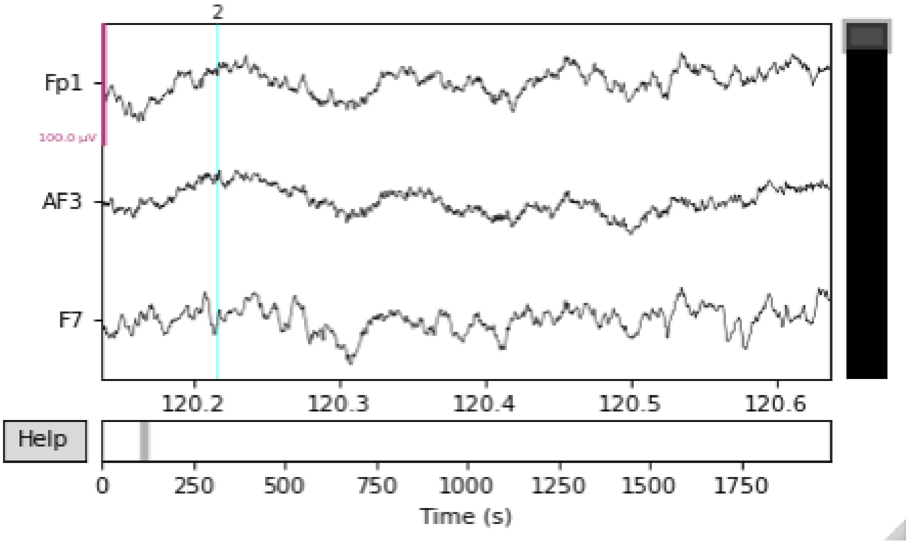

Figure 1: An EEG signal visualized with the Python MNE toolbox [12]. The electrode channels Fp1, AF3 and F7 are considered in the 100 μV range. The blue line indicates the onset of event '2'.

2.1.1 Auditory oddball task

In Figure 1 a blue line with the label '2' can be seen around the 120.2 seconds mark. This line is a marker for the onset of event '2'. Such an event can for example be a visual or auditory stimulus that is presented to the subject. The voltage change in the brain as a reaction of such an event is called the *event-related potential* (ERP). ERPs can be used to study the effect of a stimulus on the response of the evoked brain.

In auditory ERP-experiments the subjects are presented with a sequence of sounds. These sounds contains *standard* stimuli that are played frequently and *deviant* stimuli that are played less frequent. The brain's responses to the standard and the deviant are compared. This type of experiment is called the oddball paradigm. If the response to the deviant is different from the response to the standard it is assumed that the subject can distinguish the two stimuli. The response difference between a standard and deviant stimulus is called the mismatch response (MMR). In auditory ERP-experiments, the sound stimuli are often *syllables*. Infants as young as two months old already show a mismatch response between standard and deviant syllables [13]. The red line in Figure 2 is a visualisation of the MMR.

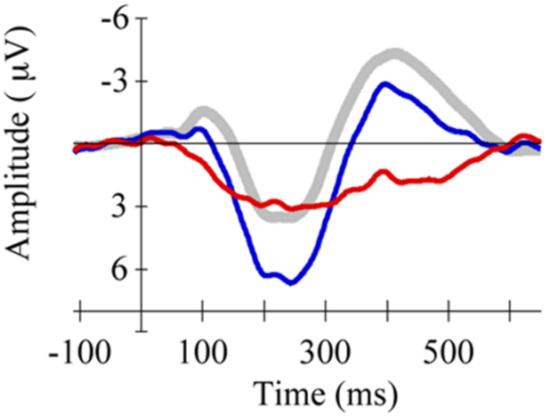

Figure 2: The mismatch response (red) as the difference between the standard (gray) and the deviant (blue) ERP response. Figure from Wanrooij et al. [13].

2.1.2 EEG research on mental disorders

EEG research is commonly used to study mental disorders including autism, depression, and epilepsy. Mental disorders are associated with irregularities in brain activity. EEG research helps to understand the causes and development of these disorders to enable better diagnosis and treatment.

Dawson et al. (2002) studied whether young children with autism have an impaired ability to recognise faces [3]. Autistic children were shown objects and faces that were either familiar or unfamiliar to them. The brain response was different between familiar and unfamiliar objects. Also, children typically have a response difference when shown familiar and unfamiliar faces. There was however no response difference in autistic children. This result can be useful for diagnosis and to better understand the limitations of autistic children.

Khodayari et al. (2013) used machine learning on EEG data to predicted whether people with major depressive disorder are responsive to the antidepressant SSRI. The overall prediction accuracy was 88% [14]. A correct prediction can help in finding suitable treatment more effectively and cost-efficient without the need to test the treatment on the subject.

Rasheed et al. (2020) reviewed research that uses machine learning to predict epileptic seizures in EEG data. Certain electrical activity in the brain can indicate an oncoming seizure [15]. Early detection of seizures is important for quick treatment with medicines or even surgery. The accuracy of models that predict seizures has been increasing over the last years due to the capability of deep learning. The network architectures that can process sequential data also become more sophisticated every year. Although deep learning provides the best predictors for seizures, it is usually unknown which patterns the network has learned from the data due to the black-box nature of these neural networks.

EEG research on dyslexia

Leppänen et al. (2011) reviewed studies to investigate whether young children with familial risk of dyslexia would process speech differently, by relating the ERPs of 'at-risk' and typical infants with the outcome of their reading ability [16]. The events in these ERPs are audio fragments that occur in speech. Children are labeled 'at-risk' of dyslexia when they have at least one dyslexic parent, since children with dyslexic parents have a higher risk of developing dyslexia themselves [9]. A difference in the ERPs of at-risk children was found between children who later become poor readers and those who became fluent readers. Leppänen et al. concludes with the statement that atypical speech processing is not likely reason by itself for dyslexia but rather a risk factor.

Van Zuijen et al (2013) found similar conclusions from an auditory oddball task. They measured whether infants at just two months could differentiate between the standard |bAk| and the deviant |dAk| syllables. They compared the MMR of typical and at-risk children. There was a difference in the MMR of typical and at-risk children that became fluent readers. Moreover, the at-risk group that later developed dyslexia did not show a MMR at all [2]. This indicates that children who become dyslexic already have an speech processing deficit well before they learn to read.

2.2 Deep Learning

Deep learning is a type of machine learning that uses a neural network to learn from data. The term *deep* refers to the hidden layers in the neural network. Multiple nodes or *neurons* in the network are structured in layers, where each layer provides an abstraction of the previous layer. Deep learning is used to discover complex patterns in tasks such as image recognition, recommendation systems, and natural language processing [17].

2.2.1 Neural network basics

An artificial neural network is a type of artificial intelligence that is modeled similarly to how a brain processes information. Neural networks are composed of a large number of interconnected *nodes* similar to neurons in the brain.

The most basic neural network is a *feedforward neural network* [18]. This is a network in which the nodes are structured in layers. The nodes between two neighboring layers are connected. A feedforward neural network is fully connected if each node is connected to every node in the next layer. The output value of each node is determined by the sum (Σ) of the input values multiplied by the weights of the node. This output is passed through an *activation function* f to introduce a non-linearity in the system. Without an activation function, neural networks could only represent linear functions. The most basic activation function is the rectified linear unit (ReLU) whose output is $f(x) = max(0,x)$.

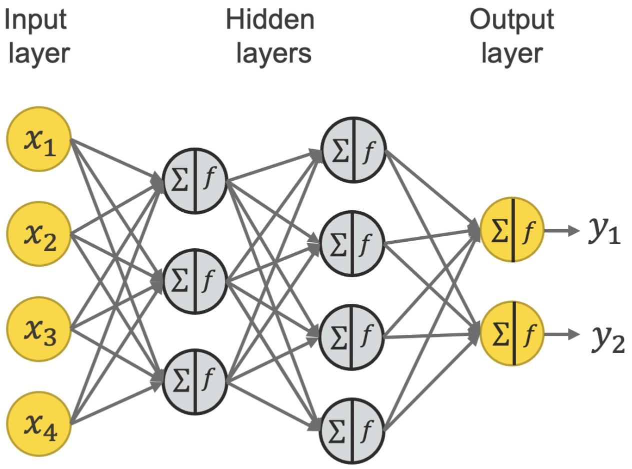

Figure 3: Feedforward fully connected neural network. In deep learning, input data such as EEG signals are passed through nodes in the hidden layers to predict output values like age and dyslexia. Figure from KNIME [19].

A neural network is trained using a process called *backpropagation*. This process adjusts the weights of the connections between the neurons in the network according to its performance on a given task. This performance is measured by making predictions on the input data and then comparing those predictions to the actual outcome. The error of these predictions is propagated back through the network to adjust the weights of the neural connections.

2.2.2 Architecture designs

Dropout is a technique in which a random subset of nodes is temporarily switched off during training. Dropout makes the network more robust, since the model will be less likely to depend on any single node. Dropout is a type of *regularization*. Regularization is a set of techniques to increase generalization with new data. Dropout makes it less likely that the model will *overfit* on the training data, which increases the accuracy to new data [20].

Convolution is a technique in which an operation sweeps over the data to extract global features. Convolution is usually followed by a *pooling* layer to reduces the feature dimensions. *Residual connections* are connections between layers that skip layers to enable deeper networks. See Figure 4 for an implementation of the aforementioned designs. In this figure the input is a single channel along the time axis. In EEG data the input usually consists of multiple channels.

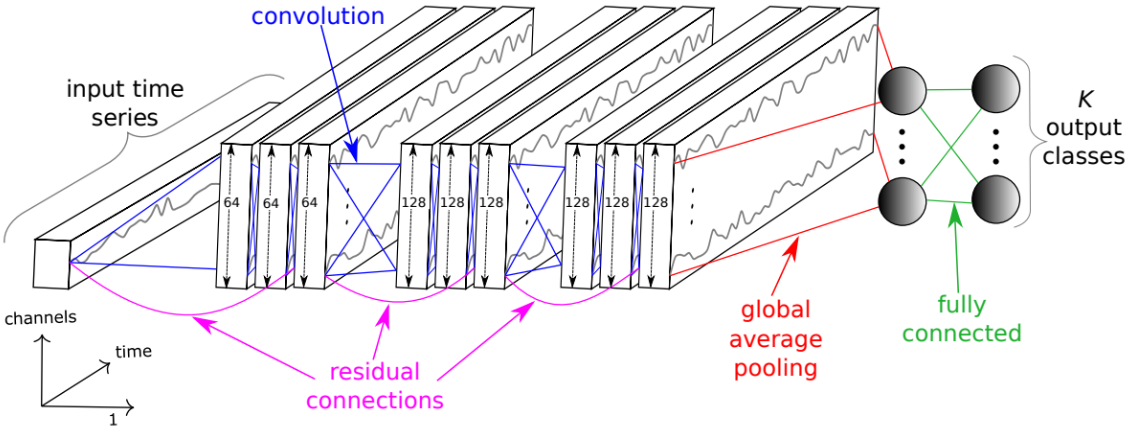

Figure 4: An architecture for time series classification with convolutional, pooling and fully connected layers. The network also contains residual connections. Note that this architecture only has one channel as input, while EEG data consists of multiple channels. Figure from Fawaz et al. [21]

Recurrent neural networks

Recurrent neural networks (RNN) are a type of neural network that are well-suited for processing time-series data. Recurrent networks have feedback connections that allow them to maintain a memory state while processing time-series data. These networks can take into account the temporal relationship between successive time-steps. This makes them well-suited for tasks where the order of the input data is important like text and EEG data processing.

The most used types of recurrent neural networks are the *long short-term memory* (LSTM) and *gated recurrent unit* (GRU). Both the LSTM and GRU are developed to manage the memory in a way that allows for long-term memory of sequential data. The drawback of these recurrent neural networks is that each calculation is dependent on the calculations of all the previous time-steps. This removes the possibility of performing parallel computations simultaneously.

Transformer

Vaswani et al. (2017) proposed the Transformer model architecture, a model that processes sequential input data such as EEG signals without using recurrence. Instead the model is based on attention [22]. For each input the model learns its significance in relation with other data, i.e. it learns where to focus its attention on. In this way parallelization is possible, which allows for much faster computation. In short, the model consists of an encoder and a decoder. The job of the encoder is to map the input data onto an abstract representation that is understandable to the model. The decoder uses this abstract representation to make predictions about the data.

Transformers have been used for classification on raw EEG data with success [23]. Siddhad et al. (2022) achieved an accuracy comparable with state-of-the-art results by classifying raw EEG data with a transformer model.

3 Available Data and Code

Before the methods of this thesis are described, it is necessary to mention the work that this thesis builds upon. The thesis would not be possible without the datasets and the many open-source repositories available in Python.

3.1 Datasets

3.1.1 ePodium (ePod)

The ePodium project developed a dataset to predict the risk of dyslexia in toddlers. In the experiment EEG-data is collected from 129 toddlers between the age of 16 and 24 months [7]. Ideally, each child performs two tests in a three month interval in which the EEG-data is recorded. The test uses the auditory oddball paradigm. The hypothesis is that children that develop dyslexia are not as skilled at distinguishing spoken syllables as typical children. If this hypothesis is true, dyslexic children can potentially be diagnosed based on an abnormality in the mismatch responses from standard and deviant syllables.

In each test the child listens to a sequence of sounds. This sequence contains 80% standard and 20% deviant syllables to elicit the mismatch response. To measure the EEG data 32 electrode channels and two mastoid references are used. The measurement frequency is 2048.0 Hz. Each test is around 30 minutes, containing four different sequences:

sequence	standard	deviant	pronunciations
1	”giep”	”gip”	1
2	”giep”	”gip”	12
3	”gop”	”goep”	1
4	”gop”	”goep”	12

For each test the vocabulary knowledge is registered with the *MacArthur-Bates Communicative Development Inventories* (CDI) questionnaire. In this questionnaire the parents fill out which words they think their child understands and which word their child uses in speech. Also, the technical reading skills of the parents are tested to determine whether they have dyslexia. The scores that determined this unofficial diagnosis are included in the data. Finally there is information on the age of the child in days at the time of the tests and whether the child is male or female.

3.1.2 Dutch Dyslexia Program (DDP)

In the *Dutch Dyslexia Program* (DDP) 300 children were followed from the age of five months up to nine years [24]. Of these children 180 have a familial risk of dyslexia (FR). In the DDP children are labeled as FR if one parent and another first-degree family member are reading impaired [25].

The EEG-signals were measured from the children every six months between 5 and 47 months. In these experiments the EEG-signals were recorded from the children, where the children listen to the dutch words ”bak” and ”dak”. Nine variation of these sounds were played, each a different combination of the two words. The event-related potentials to each distinct sound event was measured.

The program also assessed expressive and receptive language, motor development, behaviour problems, and home-literacy environment by questionnaires at the age of two and three years [26]. Speech–language and cognitive development was measured from 47 months onward. Pre-literacy and sub-skills of reading and reading development was evaluated during kindergarten and grades 2 and 3.

The bak-dak stimuli were chosen for multiple reasons [27]. In Dutch, recognising the difference between the /b/ and /d/ phoneme is more difficult for dyslexics and poor readers [28]. Moreover, /bA/ and /dA/ are already recognizable at a very young age, since these sounds are very common in the Dutch language.

The master thesis of Bruns (2021) used a subset of the DDP dataset to predict age. Data from children between the age of 11 and 47 months with six-month intervals was used [6]. Children below the age of 11 were excluded since these recordings contained a lot of noise and artifacts. It was shown that deep learning models can predict the developmental age of infants by applying deep learning to EEG data. Bruns (2021) trained both traditional machine learning and deep learning models to predict age from the data. The best model was an *encoder deep learning model.*

The main advantage of the DDP dataset is that it is very large. Deep learning is notorious for needing lots of data to be able to find patterns. The downside is that the dataset is relatively old, as the program started in the late nineties [26]. It may be more difficult to work with this data since many computing tools have changed. Beware that the following issues may occur when using the data: some files have no header, some experiments are split into multiple files, some experiments are incomplete, some age groups are larger than others, and the ’first standard’ seemed to be marked before the ’deviant’ event.

3.2 Python Code

This thesis heavily relies on software to analyse and experiment with the data. The software of the thesis is written in Python, since this programming language contains many useful packages.

3.2.1 Tools and packages

This project makes extensive use of the following well-known Python packages: *MNE*, *Tensorflow/Keras*, *NumPy*, *Matplotlib*, and *Pandas*. *MNE* is a toolbox for analyzing and visualizing EEG data. *Tensorflow* is a deep learning frameworks in which deep learning models are commonly programmed. *Keras* is built on top of Tensorflow to simplify the implementation of neural networks. *NumPy* is the standard tool for math and arrays. *Matplotlib* is the basic tool for visualizing data, and *Pandas* is a tool to load and manage data structures.

Autoreject

EEG measurements contain the neural activity of a brain, as well as unwanted electrical signals such as electrical devices and biological signals like eye movement, muscle activity, and skin potentials [11]. These noise signals are called *artifacts*. To make the data less noisy, signals with these artifacts are either rejected or corrected. Virtually all EEG signals contain some artifacts. Minor artifact should be corrected, while large artifacts that obscure most of the useful data should be rejected.

Autoreject is a Python library to automatically reject bad trials and repair bad sensors in EEG data [29]. Autoreject is dependent on the MNE package, since an *epoch* object from the MNE package is used as input. The MNE epoch object contains the EEG in 'epoched' trials surrounding the event markers, instead of as one continuous signal. The *fit_transform* method of autoreject takes the epoch object and returns cleaned epochs.

Selecting deep learning models

Fawaz et al. (2019) analysed the most promising deep neural networks on time series data [21]. The performance of each of these models is compared on multiple classification tasks. The best performing model was *ResNet*, followed by the *Fully Convolutional Network (FCN)* and the *encoder* model. The authors Fawaz et al. provide open-source implementations of the models on GitHub: `https://github.com/hfawaz/dl-4-tsc`.

Bruns (2021) used multiple traditional machine learning and deep learning models in his master thesis to make predictions on the developmental age of children in the DDP dataset [6]. The deep learning models were pulled from the repository of Fawaz et al. (2019). In the thesis the encoder model had the best performance on age prediction on the ERP signals. The encoder architecture consists of a convolutional neural network whose temporal output is summarized by a convolutional attention mechanism, to obtain a fixed-length representation from a variable-length time series [30]. The code of Bruns' thesis is available on GitHub at: `https://github.com/ePodium/EEG_age_prediction`.

Reproducibility

As this project makes use of many sources from previous work, it is important that the work of this project can also be reused in future research. Roy et al. (2019) analysed the literature on deep learning-based EEG, and noticed a high variability in how results were reported [5]. As a result, they made recommendations for researchers of *deep learning EEG analysis*. The model architecture and data should be clearly described, state-of-the-art baseline models should be included in the analysis, reproducible code including hyper-parameter choices and model weights should be shared, and the code should be able to run on another computer and on new data. This thesis makes sure that these recommendations are implemented.

This project aims at developing reusable code. The code for this project is written in Python and published in GitHub. The open-source repository can be found on `https://github.com/eegyolk-ai/eegyolk` in the *floris_thesis_code* folder. The repository contains notebooks which let the user process, visualise, and interact with the datasets. There are also notebooks to train and analyse deep learning models. The README file provides a more extensive explanation on the workings of the code. It should be possible for anyone with the datasets and an understanding of Python and EEG data to reproduce and build upon the results.

4 Methods

In this section the methods of experimentation are discussed. The results should be reproducible when using the code in the *eegyolk* repository and applying the same methods on the DDP and ePod datasets.

4.1 Preparing Data

Each dataset is set-up differently. There are many parameter differences between datasets, such as: the file extension, the set of electrode channels, the sampling rate, and the type of events in the experiment. Also, the available metadata on the participants is different for each dataset. These details need to be known before working with a dataset to prevent unexpected issues at a later stage.

The data may be incomplete. For example, not all ages of participants are known in each experiment of the DDP dataset. Issues like these could always be solved by removing the data with unknown labels. A more data efficient way is to interpolate the age from the age group of the experiment.

Both datasets have cases where a single experiment is split up into separate files. This can for example occur when the participant needs a break during measurement or the equipment needs to reset. In the code these files are recombined into a single file.

The channel montages of all experiments should be configured identically in the code to prevent compatibility errors. The ePod dataset contains 32 channels, while the DDP dataset has experiments that contain either 30 or 62 channels. 26 channels in the ePod and DDP dataset are identical. These 26 channels are used when two datasets are used in the same model. When using only the ePod or DDP datasets by themselves, the 32 and 30 channels are used respectively.

The ePodium dataset has 78 distinct events. These are reduced to 12 events, a standard, deviant, and first standard for each of the four sequences. The DDP dataset also has multiple standards and deviants in an experiment. These events are merged together, so only a single standard and deviant event type remain in the DDP dataset. Generalizing between common events simplifies the analysis, at the cost of removing details between the different event types.

4.1.1 Pre-processing steps

The data can be processed once the details of the datasets are clear and the available data is complete and structured. Processing the data is useful for removing excess noise, and to shape the data for use in the deep learning models. The entire pre-processing pipeline can be seen in Figure 5. Each individual processing step is further explained below.

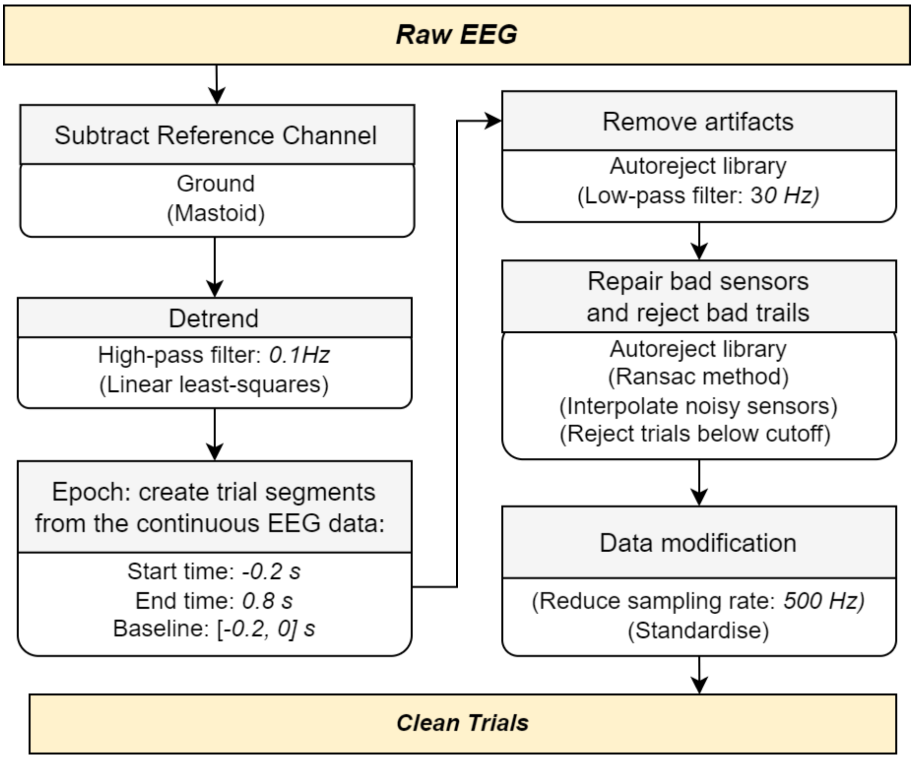

Figure 5: The EEG pre-processing pipeline from raw EEG data to clean epoched trials. Each pre-processing step is accompanied with design choices. Possible alternative options are surrounded by the round brackets.

Reference channel

Voltage is always measured relative between two points. There are no absolute values in voltage. In this project, the voltage of the EEG signals is measured relative to the ground voltage. Another reference that is often taken in EEG measurement is the average of the two mastoid channels. The mastoids are the bones behind the ear. These reference channels record roughly the same noise as the EEG channels, but they may contain more noise from the neck muscle as they are closer to the neck [11].

Filtering

EEG data is usually filtered before analysis as a way to reduce noise. The two most common filter types are the *low-pass filter* and the *high-pass filter*. The book *An introduction to the Event Related Potential Technique* by Steven J. Luck makes recommendations for the cut-off frequency of these filters. A high-pass filter of 0.1 Hz is recommended to remove voltage drifts like slow changes in skin potential. A low-pass filter of 30 Hz is recommended to reduce high-frequency signals like muscle (EMG) activity and line noise. Most of the relevant portion of the ERP waveform in a typical cognitive neuroscience experiment consists of frequencies between 0.1 Hz and 30 Hz [11]. Noise within this frequency range should not be filtered, since it overlaps with the frequencies of the relevant signals.

This project uses a high-pass filter to remove the trend in the EEG-signal. No low-pass filter is used, since the artifacts are removed with a software library that automatically corrects the artifacts within each trial.

Automatic epoch pre-processing

The continuous EEG data is epoched, which means that the signals or *trials* surrounding an event are extracted from the data. These trials are taken 0.2 seconds before to 0.8 seconds after the onset of an event. The time before the onset of the event is called the *baseline*. The average of the baseline is subtracted from the trial. This is called the baseline correction. The corrected one-second trials are stored in an *epoch* object from the MNE-Python package.

The *autoreject* algorithm takes this epoch object as input to repairs bad channel sensors and reject bad trials. Depending on the number of bad sensors, the trial is either repaired by interpolation or excluded from subsequent analysis [29]. Another automatic pre-processing algorithm is *RANSAC* (random sample consensus). This algorithm tries to correct for outliers by removing signals that behave unpredictably relative to other channels [31]. For comparison, the results of both autoreject and RANSAC on an ERP signal of the ePodium dataset can be found in the appendix. The autoreject algorithm takes longer to process all the data, but the output was generally cleaner. This is why autoreject is used to clean the data. Note that even autoreject does not remove all noise and may also remove some useful signals.

4.1.2 Data management

This project uses a shared storage in Linux. This storage is coupled to the *SURF research cloud* on which the Python code runs. The ePodium dataset consists of 103 GB of raw .bdf files, and the DDP dataset consists of 166 GB of raw .cnt files. The storage also contains the processed data, which is 127 GB for ePod and 55GB for the DDP dataset.

Processed files are stored in a MNE .fif format. The data was originally stored as a numpy array in the standard .npy format. The reasoning was that .npy files are faster to load into deep learning models than .fif files. However, epochs stored as .npy files take up much more space than .fif files. Furthermore, a .fif file contains event info, while events need to be separately stored with .npy files. Storing and loading the events separately to the data makes the code more confusing and less reproducible. In contrast, the computing time of extracting the data from the .fif files turned out to be negligible.

Data modification

One example of data modification is to add Gaussian noise to the data as a way to increase the variance in the data to prevent overfitting. Adding Gaussian noise to each data-point of a signal takes only a fraction of a second.

It is also possible to *standardise* the data. This is done by dividing the voltage values of each electrode channel by the standard deviation of the channel. The reasoning is that the signals will look more similar, which potentially increases the capacity to generalise between different data inputs.

With *downsampling* the temporal resolution is reduced. The ePod dataset is sampled at 2048 Hz, which means there are more than two data-points for each millisecond. Frequencies above 30 Hz are often filtered, which makes it unlikely that useful information remains within the millisecond range. Small differences between data-points are also averaged out when the trials are averaged into an ERP waveform. An added advantage to downsampling is that the data size is reduced.

The efficiency of loading the data is very important, since a deep learning model loops over the same data multiple times. Downsampling takes a relatively long time to process, so it is impractical to downsample the original data each time the data is loaded. It is preferred to have a separate downsampled dataset stored during training and testing.

4.2 Implementing Deep Learning

At this stage, the clean data has been stored in a drive that is accessible from a Python script. In this section multiple models and settings are selected that optimally learn and predict patterns from the stored data. The data is shaped so it can act as input for deep learning models. The deep learning experiments that are performed in this thesis are now described in more detail.

4.2.1 Model set-up

In Figure 6 the model settings along with the possible values are visualized. These settings are further explained below.

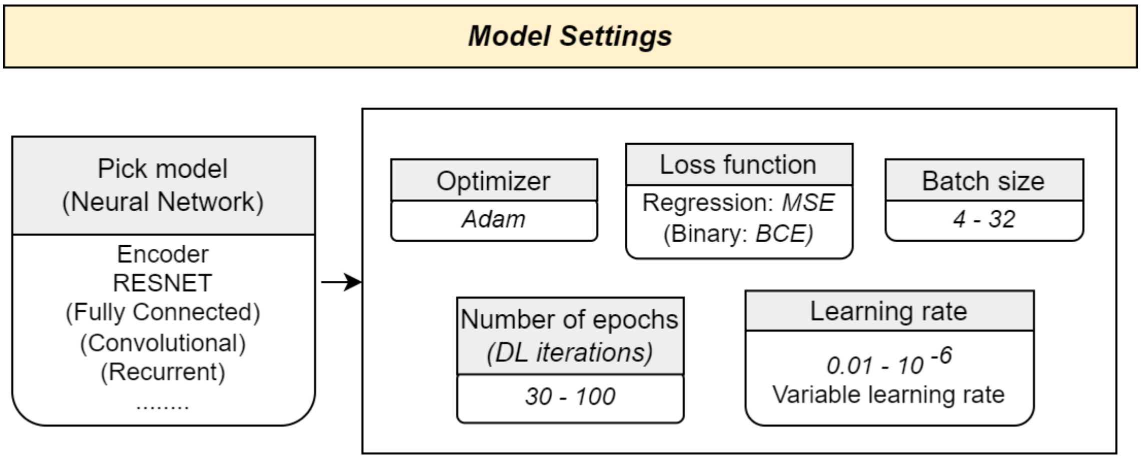

Figure 6: Deep learning models along with a range of model settings.

In deep learning, the error of a prediction is called the *loss*. One way to calculate the loss is to use the *mean squared error* (MSE) loss function. This function calculates the average of the squared differences between the estimated and the actual values. With MSE an estimation that is twice as accurate reduces the loss by a factor of four, and vice versa.

The *optimizer* of a deep learning model uses the loss to update the weights of the neural network. The goal of the optimizer is to minimize the loss of the model predictions. This is done by calculating the *gradient* of the loss with respect to the model weights. In other words, the optimizer calculates changes to the loss as a result of changes in the model weights. The weights are subsequently adjusted in the direction of lower loss.

The models of this project use the popular *Adam* optimizer [32]. This optimizer works on noisy data, has a fast convergence rate, is computationally inexpensive to implement, and is well-suited to a wide range of optimization problems.

The magnitude of the adjustments to the weights is called the *step size*. A high step size results in faster learning, but adds the risk of overshooting the optimal weight settings. A low step size takes longer to train but can result in a more optimal model. The step size is related to the *learning rate* of the model. A learning rate ranges from values between 10^{-6} and 1. In this project the learning rate is variable. The learning rate is initially high and is reduced when the validation loss no longer improves.

During training, the model iterates over the entire training set multiple times. These iterations are called *epochs*. In each epoch, multiple data samples are passed through the network simultaneously in a *batch*. The model weights are updated after processing a batch of data. Training in batches has the advantage that the calculated loss has less variance. Training is also faster, since the model is only updated after processing a batch instead of after each instance.

4.2.2 Shaping input data

Deep learning models can learn from virtually any data that contains patterns. The models use the input data of ERPs from standard events in most of the experiments in this thesis. Some models that predict the risk of dyslexia also use the mismatch response, since this signal is expected to show signs of dyslexia. Other types of input into a deep learning model are for example raw EEG segments, Fourier transforms of EEG signals, or a set of features that describe the data. These types of input are not used by any model in this thesis. In Figure 7 the pipeline from data to model inputs is shown.

In an ideal situation, each experiment of both the ePodium and DDP dataset contain 360 trials with a *standard* stimulus excluding the *first standards*. There are four sequences in both sets, giving a total of 1440 standards. In the ePodium dataset one in four stimuli is a deviant, and in a DDP experiment this is one in ten. The cleaned experimental data rarely contains all the 1440 standard trials due to trial rejections and a possible early termination of the experiment. In the appendix a histogram of the number of remaining valid trials can be seen for both experiments.

Experiments with few remaining trials are removed from the input data. Only the standards are used from the DDP dataset. In this set the experiments with fewer than 180 standards are excluded. This arbitrary boundary is chosen since 180 is half of the 360 standards present in any sequence and 180 trials are sufficient to create an ERP. With the ePodium dataset the minimum requirement is 180 standards and 80 deviants in each sequence. A minimum number of deviants is necessary because some experiments use the mismatch response in the ePodium dataset. 186 out of 248 experiments satisfy these conditions.

The dataset is split into a *training*, *validation*, and *testing* set. The model learns from the training set by adjusting the model weights in the direction of the correct target outputs. The validation set is used to validate the model performance during training and pick the model with the lowest loss. The test set is used for an unbiased evaluation of the performance of the final model. The training set is usually the largest of the three sets. The participants are randomly inserted in one of the sets, and each model has a different combination of participants in each set.

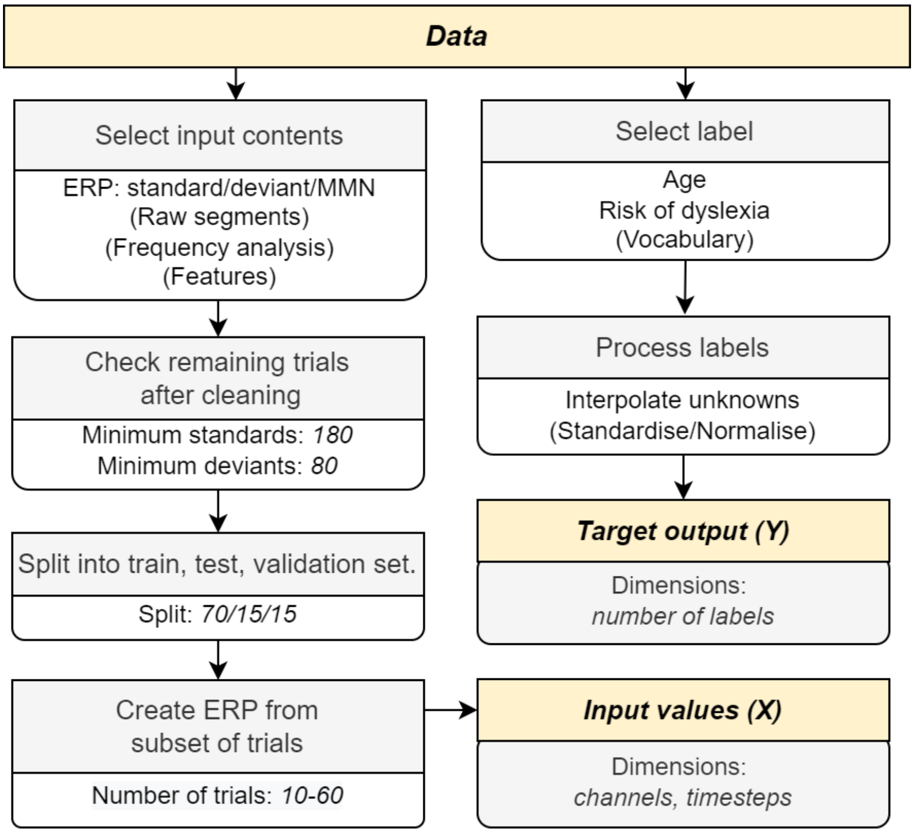

Figure 7: The pipeline for creating input data to train deep learning models. The input values (X) are ERPs, and the target output (Y) is the label the model learns to predict.

ERPs from averaged trials

An ERP is a measured brain response to a stimulus. ERPs are collected by taking the average signal of multiple trials from the same event. The number of trials required for an ERP depends on several factors, such as the size of the ERP effect being examined, and the amplitude of unrelated activity [11]. The necessary trials to create an ERP ranges anywhere from 10 to 500 trials for each event.

In Figure 8 the procedure from trials to an ERP is visualised. ERPs are created by taking the average of a random subset of trials. These ERPs are used for training the models to predict the target output. In this thesis the target outputs are either *age* or the *risk of dyslexia* depending on the experiment.

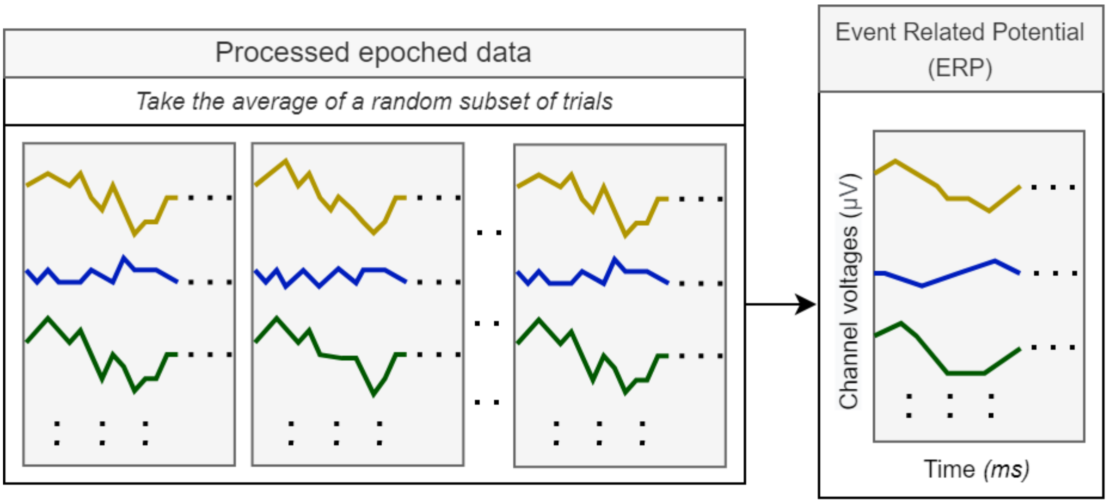
Processed epoched data
Take the average of a random subset of trials
Event Related Potential (ERP)
Channel voltages (µV)
Time (ms)

Figure 8: ERP data from the average of sampled trials as input for training a model.

4.3 Experiments

Experiments on DDP

The first experiment in this thesis is to reproduce the results from the master thesis of Bruns (2021). The DDP dataset is trained on the encoder model, as well as on the ResNet model. These models are pulled from the GitHub page of Fawaz et al. (2019). According to this source the ResNet model has the best performance on the most time-series classification tasks [21], while the encoder ranks third.

In Bruns (2021) the best performing model was the encoder model with a *mean absolute error* (MAE) of 4.82 months and a *coefficient of determination* (R^2) of 0.674 with 7-fold cross validation on the entire DDP dataset [6]. Without cross-validation the encoder model reached a MAE of 5.06 and an R^2 of 0.613. ResNet predicted the age with a MAE of 5.78 and an R^2 of 0.490 on the DDP dataset.

Bruns (2021) used ERPs as input for the age prediction models. To create these ERP signals, 30 random standard trials were sampled and averaged for every experiment at each training epoch. One microvolt of Gaussian noise was added to each signal, and the signals were normalised. These settings are also used to reproduce the results.

After the results are reproduced, the encoder model is trained with different settings for the input ERP, and the performance differences are compared. The model is trained without the standardisation to see if standardisation increases performance. Other settings that are modified are the number of trials that make up an ERP and the standard deviation of the artificially added Gaussian noise. The results may give a better insight in more optimal parameters.

Experiments on ePod

The models trained on the DDP dataset will be tested to make predictions on the ePodium dataset to study if and to what extend the models are transferable between datasets. The best performing models on both the standardised and non-standardised data of the DDP dataset are used to make predictions on the ePodium dataset.

The models that are trained on the DDP dataset are also trained further on the ePodium dataset to predict age. The performance of these models are compared with the performance of the models that are trained on ePodium from scratch, to investigate whether training a model on multiple datasets increases the performance.

Finally the risk of dyslexia is predicted. The ePodium dataset does not contain information on whether the children will develop dyslexia. There are however scores of the parents on three dyslexia tests. These tests are EMT, Klepel, and verbal competence on the WAIS test. If both parents did the tests there are six scores per participant, and three otherwise. The scores to each of these tests are normalised between 0 and 1, with 0 the lowest score and 1 the highest score in the dataset. These normalised test scores are averaged together to obtain the final value for *risk of dyslexia*. See Figure 9 for the distribution of these scores in the ePodium dataset.

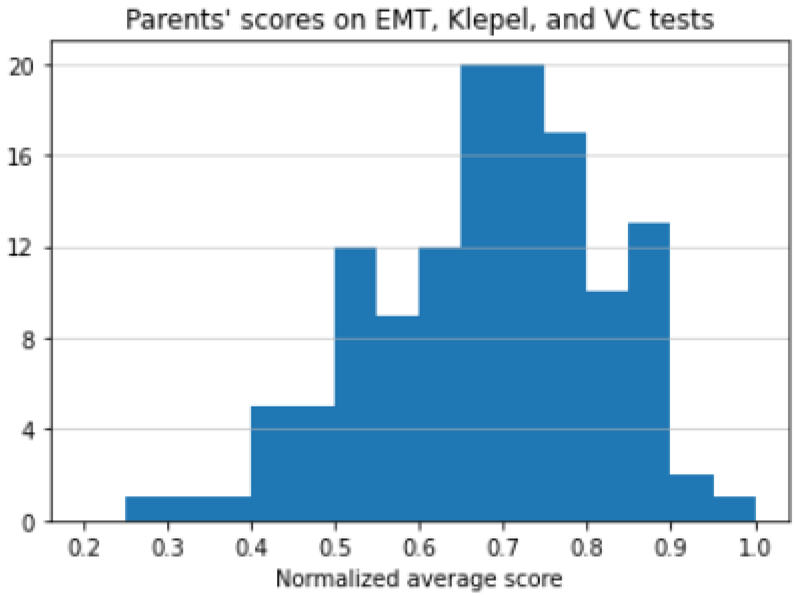

Figure 9: Distribution of the average normalised scores of the parents on the dyslexia tests in the ePodium dataset.

5 Results and Discussion

5.1 Reproducing results

The encoder model is trained five times from scratch on the DDP dataset to predict the age of the participants. The input ERPs consist of the average of 30 randomly sampled standard trials for every experiment at each training epoch. One microvolt of Gaussian noise is added to each signal, and the signals are normalised. The model is trained on the training set, the best model is chosen with the validation set, and the performance of the model is measured with the test set. These performance results can be seen in Table 1.

	MAE	RMSE	R^2
1	5.56	6.61	0.526
2	7.07	7.72	0.438
3	6.41	7.66	0.456
4	5.07	6.17	0.501
5	5.38	6.42	0.460

Table 1: The *mean absolute error* (MAE), *root mean squared error* (RMSE) and the coefficient of determination (R^2) from five encoder models with the same settings.

For MAE and RMSE lower values are better, while a higher value is better for R^2. Model 4 has the lowest MAE and RMSE, and model 1 has the highest R^2, as can be seen from Table 1. Model 4 has a MAE of 5.07 and model 1 has a R^2 of 0.526. The MAE is close to the results of Bruns (2019) where a single encoder model reached a MAE of 5.06 with an R^2 of 0.613.

As can be seen from Figure 10 there is a clear correlation between the actual age and the predicted age. There are however no predictions under 16 and above 35 months, while the ages range from 10 to 47 months. This is because the model tries to minimize the loss, and the loss is on average lowest near the mean of the dataset. Since the model is never 100% confident, the predictions have a bias towards the mean.

The experiments in Figure 10 all have a distinct color. This indicates the number of remaining standard trials in each experiment. Experiments where more trials are rejected generally contain more noise. Noise can be caused by movement from the participant. A hypothesis is that younger children have a more difficult time to remain motionless, in which case the model may predict the age from artifacts caused by movement instead of

EEG signals originating from the brain. Figure 10 does not support this hypothesis that noisy (red) experiments have a lower age prediction. In fact, the figure seems to slightly indicate the opposite.

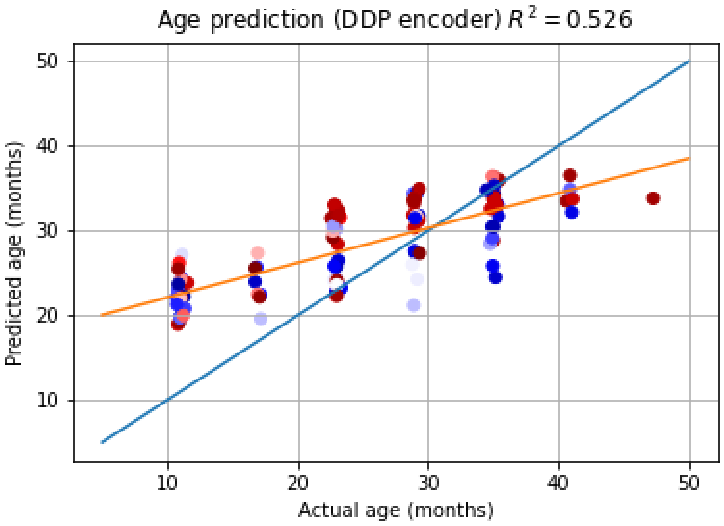

Figure 10: Scatter plot of the predicted age compared to the actual age of from encoder model 1. The colors indicate the amount of standard trials in an experiment, ranging from 180 to around 2000 trials, where the red color indicates fewer trials.

ResNet

In addition to the encoder, the ResNet model is also trained on the DDP dataset. The other settings are identical in both model types. The performance of five trained models can be seen in Table 2. Model 3 and 4 have the best performance comparable to the performance of the ResNet model from Bruns (2019) with a MAE of 5.78 and an R^2 of 0.490 on the ResNet model.

	MAE	RMSE	R^2
1	6.12	7.44	0.396
2	6.05	7.34	0.413
3	5.54	6.54	0.440
4	5.53	6.55	0.437
5	6.18	7.44	0.363

Table 2: Results from five ResNet models with the same settings.

Age predictions based on multiple ERPs

To create an ERP as input to the model, 30 trials are randomly sampled from the standard trials of an experiment. These trials are averaged and the age is predicted based on the resulting ERP. Since the ERP from an experiment is different each time, the age prediction of a model on the same experiment will also be different. If the age prediction is based on just a single ERP, only a small subset of the data in each experiment is used for prediction. This is why multiple ERPs are used for prediction, where each ERP is created from a different subset of 30 random trials. These predictions are averaged to obtain the final age prediction.

The number of ERPs used for prediction is plotted against the MAE in Figure 11. These predictions are made by encoder model 4. Taking the average of multiple predictions based on multiple ERPs reduces the MAE, as can be seen in the figure. The variance in the MAE is also reduced with an increasing number of predictions.

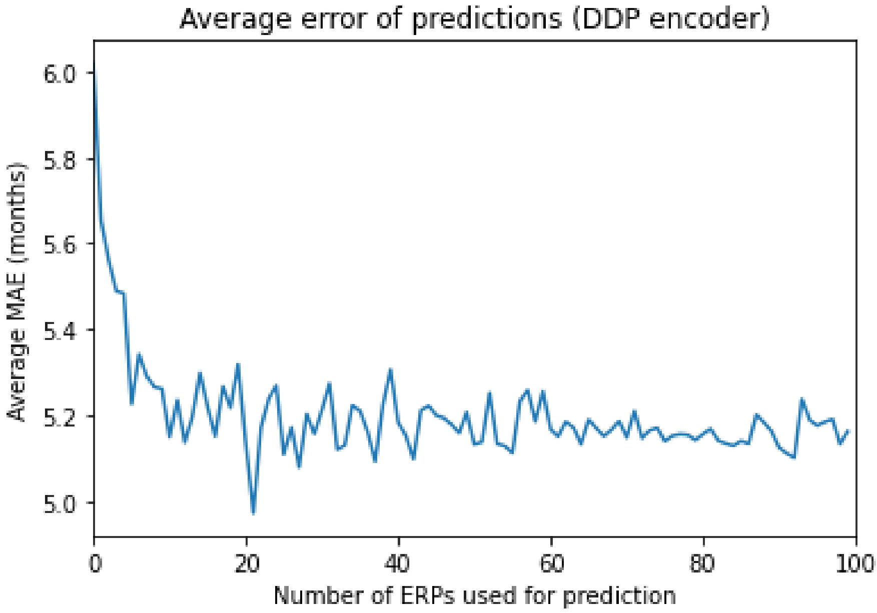

Figure 11: The average MAE compared to the number of ERPs to make a prediction on each experiment in the test-set.

5.2 Input tuning

No standardisation

Up until now the ERPs are standardised, meaning that each individual voltage measurement in the ERP is divided by the standard deviation of all the data-points in the ERP. The encoder model from Table 1 is once again trained with 1 μV of Gaussian noise and 30 trials averaged to create the ERPs. However, the data is not standardised this time. The results are shown in Table 3.

	MAE	RMSE	R^2
1	6.21	7.69	0.452
2	5.65	7.16	0.422
3	5.97	7.27	0.489
4	6.12	7.48	0.423
5	6.40	7.94	0.400

Table 3: Encoder performance without standardisation.

Some observations can be made by comparing Table 1 using standardised data with Table 3 using non-standardised data. Table 1 contains the best performing model, but also the worst performing model. The performances of Table 3 are closer together. According to the results in the tables, standardised data as input results in more variance in the performance of the models. The reason for this is unknown, as it was expected that the variance of the results would decrease with standardisation instead of increase

Trial averages and noise

The other inputs that are varied to compare model performances are the number of trial averages and the Gaussian noise. These models are not standardised since standardisation seemed to increase the variance in the results.

In the appendix can be seen how the number of trial averages affects the ERP. The ERP in Figure A5 is the average of 15 trials, while the ERP in Figure A6 is the average of 60 trials. The amount of variance and inherent noise is lower in signals with more number of trials, while the global pattern of the signal can already be identified from the average of only a few trials.

The appendix also contains a single ERP channel that does not contain artificial noise in Figure A7, and 1 μV of Gaussian noise added to the channel in Figure A8.

	MAE	RMSE	R^2
15-0	6.70	8.27	0.331
15-2	6.08	7.64	0.436
45-0	6.51	8.20	0.438
45-2	5.53	6.72	0.584
60-0	5.07	6.32	0.637
60-2	5.90	7.11	0.453

Table 4: The performance of models that differ in the number of trials in an ERP and the strength of Gaussian noise in microvolt. For example the experiment 60-2 averages 60 trials for an ERP and contains 2 μV of artificial Gaussian noise.

From Table 4 can be seen that the artificial noise increases performance when few trials are averaged, and reduces performance when many trials are averaged. This may be due to the nature of ERPs with few trials, which have a higher noise and variance by default. The model may learn to ignore this noise when artificial noise is present as well. In the ERPs with many trials there is less noise and a lower variance, so the only effect of artificial noise in these cases may be the inherent information loss from adding noise.

It is interesting to note that the model performance never drastically decreases when $2\mu V$ of noise is added to the data. Even $1\mu V$ has a clear effect on the data as can be seen in Figure A8. The steady performance with noisy input data is an indication that the age is inferred from the global pattern of the ERPs instead of the point-to-point differences in time, as the noise is added independently to each time-point.

It should be noted that the models '45-0' and '60-0' are similar in model input but still have very different results. This indicates a high variance of the model performance. The models are only trained once on each setting, which means that no definitive conclusion can be made from these results.

More averages seem to have a better performance than few. This is an indication that the model predicts the age on the basis of the response to an event, as the ERP is more clear when more trials are averaged. This may also explain why the model can already predict the age fairly optimal with only a few ERPs. A prediction near the optimal prediction can be made with only a fraction of the total available standards from an experiment, as can be seen from Figure 11. This may be because this fraction of trials is enough to make an ERP that contains most of the information needed to predict age.

5.3 Predictions on ePodium dataset

5.3.1 Transfer learning

The encoder models that are trained on the DDP dataset are now applied on the ePodium dataset. These models are the same five models that used the standardised ERPs from the DDP dataset to make the results of Table 1. The performance from these models applied on the ePodium dataset is shown in Table 5.

	MAE	RMSE	R^2
1	2.36	2.84	-0.668
2	2.08	2.53	-0.323
3	1.97	2.44	-0.262
4	2.35	2.86	-0.686
5	2.11	2.57	-0.370

Table 5: Performance of the encoder models trained on standardised ERPs from the DDP dataset, and applied onto the ePodium dataset.

Note that the MAE and the RMSE of the models are lower on the ePodium dataset than on the DDP dataset. This does not mean that the model performs better on the ePodium dataset, but is mainly because the age range of the participants is lower in the ePodium dataset. In general MAE and RMSE values are not comparable between different datasets, due to a different age distribution of the participants.

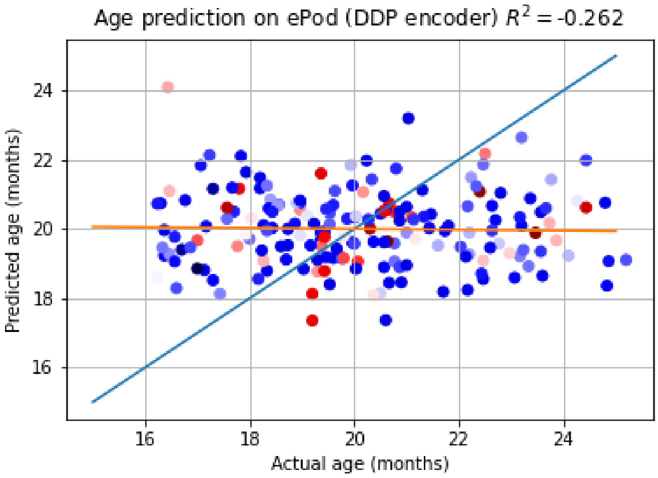

Figure 12: The age predictions on the ePodium dataset from model 3 of Table 5. The colors indicate the amount of standard trials in an experiment, ranging from 1100 to around 1536 trials, where the red color indicates fewer trials.

As can be seen from Figure 12, the model roughly predicts the average age of the participants. The model predicts on average roughly 20 months on the ePodium dataset even though the average of the DDP dataset is around 26 months. The coefficient of determination (R^2) is however negative in each model, which is worse than a model that would always predicts the mean of the data.

Now the encoder models trained on the non-standardised DDP data are applied onto non-standardised ePodium data.

	MAE	RMSE	R^2
1	5.68	6.20	-6.956
2	4.31	4.93	-4.030
3	4.94	5.46	-5.163
4	3.69	4.24	-2.722
5	4.77	5.30	-4.804

Table 6: Performance of the encoder models trained on non-standardised ERPs from the DDP dataset and applied onto the ePodium dataset.

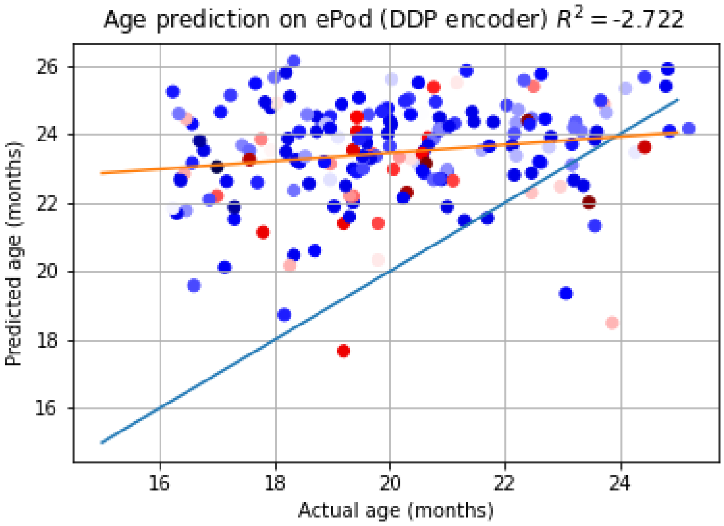

Figure 13: The age predictions on the ePodium dataset from model 4 of Table 6. The colors again indicate the amount of standard trials in an experiment, ranging from 1100 to around 1536 trials.

By comparing Table 6 with the standardised results of Table 5 it is clear that the performance of the standardised data is much better than with the non-standardised data. Predicted ages on the non-standardised ePodium dataset are much closer to the mean of the DDP dataset. However, the non-standardised data has a positive slope in the regression line of the predicted and actual age, which is not the case for the standardised data. There seems to be a trade-off between finding the mean of the dataset as a whole and a correlation within the dataset between standardised and non-standardised data. It is unclear why this is the case.

It is notable that the models trained on DDP are somewhat applicable to the ePodium dataset even though the experiments of the datasets have many differences. For example different syllables are used as the auditory stimulus.

Now the encoder model is trained on age on the ePodium dataset. This is done for the standardised and non-standardised data. The model are trained from scratch, as well as with transfer learning where the weights of the models trained on the DDP dataset are used as initial values.

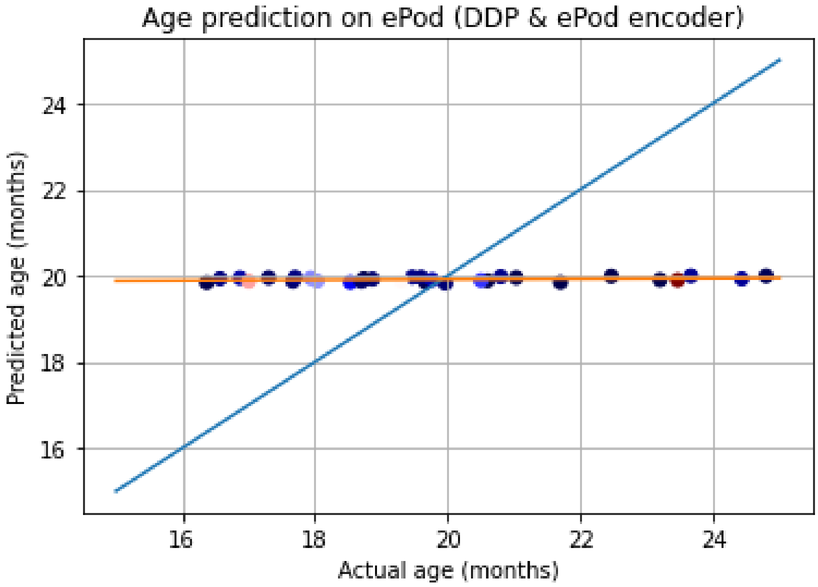

Figure 14: The age prediction of a model trained on the ePodium dataset with a RMSE of 2.34 and a MAE of 1.93 months.

The model in Figure 14 has not found any correlation between the ERPs and age of the participants. A similar result is found in each model that is trained lastly on the ePodium dataset for both standardised and non-standardised ERPs.

The predicted value is always near the mean. This may be due to the small range of ages in the participants or the relatively few experiments compared to the DDP dataset.

5.3.2 Risk of dyslexia

Now the risk of dyslexia is predicted. The input is the difference between the standard and the deviant response as it is expected that children at-risk of dyslexia have a less pronounced MMR. An average of 80 standards and deviants are used to create the MMR. The model attempts to predict the average of the parents' scores on dyslexia tests based on the MMR.

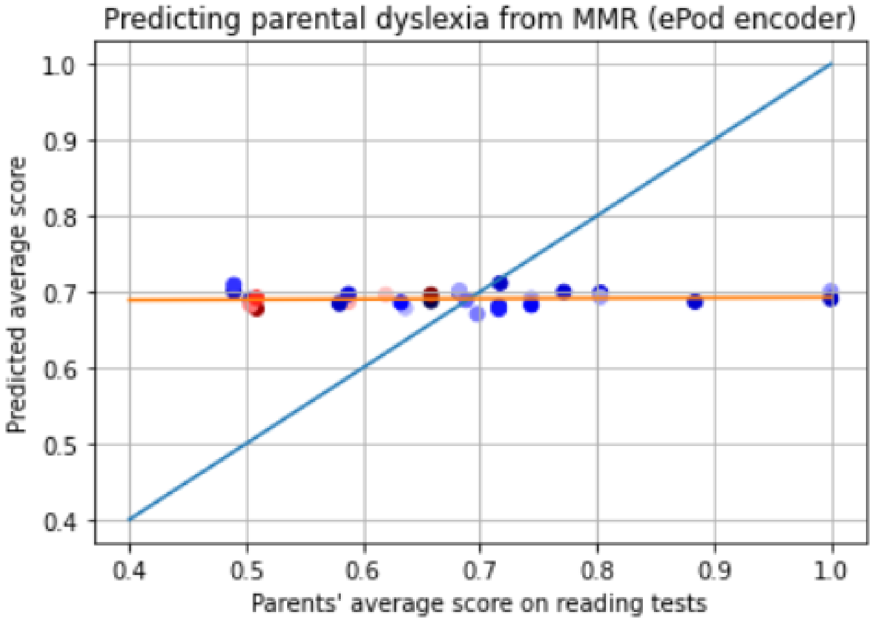

Figure 15: Encoder model predictions of the averaged dyslexia scores of the parents from the MMR in the ePodium dataset.

As can be seen from Figure 15 the encoder model did not find any correlation between the MMR and the parental scores of dyslexia in the participants. It may be that these average scores are a bad indication for the risk of dyslexia. Other reasons why the model did not find any correlation may be due to a small training set, or because the effect that risk of dyslexia has on the MMR is only small and difficult for the model to observe.

6 Further Research

A way to improve upon the results of this thesis is to apply the methods to *more datasets*. There are many open source EEG-datasets. For example, the GitHub page `https://github.com/meagmohit/EEG-Datasets` contains a list with over a hundred links to public EEG-datasets. Some of the datasets are in the ERP oddball paradigm. A few datasets have an auditory experiment, but there are no datasets to evoke signs of dyslexia.

There are multiple advantages of using multiple datasets to train a single model. The model can train on more data, which results in more accurate results. The model is also able to make predictions on multiple models. However, the difficult aspect of creating a model that is transferable between datasets is to deal with the many differences between each dataset, like the different types of experiment, available metadata, channel montages, sampling rates, and measuring devices.

Another way to improve upon the results is to compare a wider range of *input settings*. The number of trials to average for an ERP and the artificial Gaussian noise could be varied further to test the model performances in the limit. Similarly, the models could be compared for different sampling rates to examine what temporal resolution is needed at minimum to predict age from ERPs.

In addition to modifying the input settings, the model performance could also be compared by varying the *model settings*. These settings may include the chosen optimizer, loss function, batch size, learning rate, and number of epochs.

Additional model architectures like the *Transformer architecture* could be used to improve upon the results. GitHub contains multiple repositories that contain a Transformer model that can handle EEG-data as input. The *EEG-Transformer* repository (`https://github.com/eeyhsong/EEG-Transformer`) contains a transformer model able to classify EEG-data on multiple categories, reaching state-of-the-art performance with fewer parameters than previous methods [33]. This model contains an attention mechanism to perceive the most relevant spatial and temporal features of EEG signals, similar to the encoder model. TensorFlow also has a publicly available transformer model in its EEG-DL library (`https://github.com/SuperBruceJia/EEG-DL`). Both the repositories are published very recently and show great potential due to their accessibility and documentation. Due to time constraints, the Transformer model is not yet properly implemented into the *eegyolk* repository.

In this thesis ERP data is the only input that is used to train the models, but there are more possible *input types* possible from the raw EEG data for deep learning. In traditional EEG analysis, spectral features such as the mean, variance, and peak voltage are often used as input in predictive models. However, raw EEG data has been shown to outperform these features in deep learning model [34], since deep learning models create their own feature representation from the EEG data.

Another type of widely used input is to use *time-frequency analysis*. In this analysis the Fourier transform of the EEG signal is used to obtain the frequencies within time signals. The frequency spectrum provides additional information about neural synchrony that is not apparent in the EEG signal [35]. One important fact to keep in mind is that the frequencies of the Fourier transform do not necessarily correspond with the frequencies of the brain waves, and vice versa [11].

When a deep learning model is unable to find a pattern in the data, there is often too little data compared to the number of input dimensions in the data. A way to *reduce the dimensions* is to use *Independent Component Analysis* (ICA). This technique reduces the dimensions while still keeping as much information within the data as possible [36].

When data is sparse, it is also possible to increase the data by using *simulated data*. Using simulated data similar to real data for pre-training models can increase performance in deep learning [37]. Figure A12 in the appendix contains a simulated EEG signal.

The model needs to be accurate as well as *explainable* if the predictions of a deep learning model are used in the medical sector. Deep learning models are hard to interpret due to their complex structures. Luckily, there are methods to get a better view of their inner workings. As part of increasing explainability in deep learning Bruns (2019) looked at the weights of a fully connected model to see which time-steps and which channels had the most influence on the model prediction. The better performing encoder model is however more difficult to interpret due to its more complex architecture [30]. It may still be possible to determine how much attention the model gives to different parts of the data by perturbing the data and measuring the change in the model predictions [38]. Since encoders are an attention-based model, it may also be possible to look under the hood to find which time intervals and channels the encoder regards as most important.

7 Conclusion

This thesis uses the DDP and ePodium datasets to make predictions on the age and risk of dyslexia of young children from EEG data. The data is pre-processed with the autoreject library, which is able to remove many of the artifacts in the EEG signals. The cleaned trials are averaged to create ERPs. These ERPs are used by deep learning models to predict the age and risk of dyslexia from patterns in the data.

This thesis has been successful at reproducing the results of the previous master thesis affiliated with the ePodium project. Both the encoder and ResNet deep learning models found comparable results on the DDP dataset. The encoder is the best model for predicting age with a mean absolute error of 5.07 months. The participants in DDP are between 11 and 47 months old and the model makes its predictions between 15 and 35 months of age.

The models can already make reasonable predictions on the age with ERPs from a small subset of the total standard trials. The models do not require every standard trial from an experiment to gain insight in the general response to the standard event. Adding a significant amount of Gaussian noise to each ERP signal does not significantly alter the performance of the models. This indicates that the models make their predictions from the global pattern of the ERP and not from local voltage differences in the millisecond range.

The models that are trained on the DDP dataset can be used to make predictions on the ePodium dataset. The models that trained on standardised data of the DDP were good at predicting the average age from the standardised ePodium dataset, while the non-standardised models found an age correlation within the ePodium dataset. It is still unclear why models that use either standardised and non-standardised data have such opposite predictive behavior.

The models were unable to find patterns for predicting age and dyslexia from the ePodium dataset alone. Some options for further research are to use more datasets with transfer learning, explore more variations in input and model settings, implement the Transformer model, use alternative input types such as time-frequency analysis, use ICA to reduce the input dimensions, use simulated data to artificially increase the size of the dataset, and make the models more explainable.

References

[1] M. Lovett, J. Frijters, M. Wolf, K. A. Steinbach, R. Sevcik, and R. D. Morris. *Early intervention for children at risk for reading disabilities: The impact of grade at intervention and individual differences on intervention outcomes (2016).* Journal of Educational, Psychology.1037/edu0000181.

[2] T.L. van Zuijen, A. Plakas, B.A. Maassen, N.M. Maurits, and A. van der Leij. *Infant ERPs separate children at risk of dyslexia who become good readers from those who become poor readers. (2013).* Dev Sci. 16(4):554-563.

[3] G. Dawson, L. Carver, A. N. Meltzoff, H. Panagiotides, J. McPartland, and S. J. Webb. *Neural correlates of face and object recognition in young children with autism spectrum disorder, developmental delay, and typical development (2002).* Child Development, 73 , 700 – 717.

[4] C. Barros, B. Roach, J. M. Ford, A. P. Pinheiro, and C. A. Silva. *From sound perception to automatic detection of schizophrenia: An EEG-Based deep learning approach (2022).* Frontiers in Psychiatry.

[5] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert. *Deep learning-based electroencephalography analysis: a systematic review (2019).* Journal of Neural Engineering.

[6] B. M. A. Bruns. *Predicting developmental age in young children by applying deep learning approaches to EEG data (2021).* Master Thesis, Utrecht University, supervisors: Dr. F. Huber, Dr. H.G. Schnack.

[7] K. Wanrooij et al. *Project ePodium: New dataset from longitudinal study on oddball paradigm (EEG-data from toddlers) (2022).* Utrecht University, UMC Utrecht, eScience Center Amsterdam.

[8] G. Stefanics, T. Fosker, M. Huss, N. Mead, D. Szucs, and U. Goswami. *Auditory sensory deficits in developmental dyslexia: a longitudinal ERP study. (2011).* 57(3):723-732. doi:10.1016/j.neuroimage.2011.04.005.

[9] T.S. Scerri and G. Schulte-Körne. *Genetics of developmental dyslexia (2010).* European Child & Adolescent Psychiatry 19, 179–197.

[10] N. M. Raschle, M. Chang, and N. Gaab. *Structural brain alterations associated with dyslexia predate reading onset. (2011).* Neuroimage, 57(3), 742-749. doi: 10.1016/j.neuroimage.2010.09.055.

[11] S. J. Luck. *An introduction to the event-related potential technique (2014)*. second edition, MIT press.

[12] A. Gramfort, M. Luessi, E. Larson, D. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, and M. Hämäläinen. *MEG and EEG Data Analysis with MNE-Python (2013)*. Frontiers in Neuroscience 267, 7, 1-13.

[13] K. Wanrooij, P. Boersma, and T.L. van Zuijen. *Fast phonetic learning occurs already in 2-to-3-month old infants: an ERP study (2014)*. Frontiers in Psychiatry, doi: 10.3389/fpsyg.2014.00077.

[14] A. Khodayari-Rostamabad, J. P. Reilly, G. M. Hasey, H. de Bruin, and D. J. MacCrimmon. *A machine learning approach using EEG data to predict response to SSRI treatment for major depressive disorder (2013)*. Clinical Neurophysiology volume 124 issue 10, 1975-1985.

[15] K. Rasheed, Q. Khansa, A. Qayyum, J. Qadir, S. Sivathamboo, P. Kwan, L. Kuhlmann, T. O'Brien, and A. Razi. *Machine Learning for Predicting Epileptic Seizures Using EEG Signals: A Review (2020)*. IEEE Reviews in Biomedical Engineering 14, 139-155.

[16] P. H. T. Leppänen, J. A. Hämäläinen, T. K. Guttorm, K. M. Eklund, H. Salminen, A. Tanskanen, and H.Lyytinen. *Infant brain responses associated with reading-related skills before school and at school age. (2012)*. Neurophysiologie Clinique/Clinical Neurophysiology.

[17] Wikipedia - Deep learning. *https://en.wikipedia.org/wiki/Deep_learning*. Accessed 14 November 2022.

[18] Jürgen Schmidhuber. *Deep learning in neural networks: An overview (2015)*. Neural Networks. 61: 85–117. arXiv:1404.7828.

[19] K. Melcher. A Friendly Introduction to [Deep] Neural Networks (2022). *https://www.knime.com/blog/a-friendly-introduction-to-deep-neural-networks*. KNIME, Accessed 9 April 2022.

[20] A. Géron. *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems (2019)*. second edition. O'Reilly.

[21] H.I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.A. Muller. *Deep learning for time series classification: a review (2019)*. Data Mining and Knowledge Discovery, 33, 4, 917–963.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. *Attention is all you need (2017)*. Advances in neural information processing systems 30.

[23] G. Siddhad, A. Gupta, D. P. Dogra, and P. P. Roy. *Efficacy of Transformer Networks for Classification of Raw EEG Data (2022)*. arXiv: 2202.05170.

[24] B. Maassen. *Early precursors of familial dyslexia: A prospective longitudinal study (2001)*. https://easy.dans.knaw.nl/ui/datasets/id/easy-dataset:112935 University of Groningen, DANS - Data Archiving and Networking Services.

[25] A. Chen, F. Wijnen, C. Koster, and H. Schnack. *Individualized early prediction of familial risk of dyslexia: A study of infant vocabulary development (2017)*. Frontiers in Psychiatry.

[26] A. van der Leij, E. van Bergen, T. van Zuijen, P. de Jong, N. Maurits, and B. Maassen. *Precursors of developmental dyslexia: an overview of the longitudinal Dutch dyslexia programme study (2013)*. Dyslexia 19.4: 191-213.

[27] F. van Beinum, C. Schwippert, P. Been, T. Van Leeuwen, and C. Kuijpers. *Development and application of a/bAk/–/dAk/continuum for testing auditory perception within the Dutch longitudinal dyslexia study (2005)*. Elsevier, Speech Communication 47, 124-142.

[28] R.S. Irausquin. *Quality and use of phonological representation in poor and normal readers (1997)*. Doctoral Thesis.

[29] M. Jas, D. Engemann, Y. Bekhti, F. Raimondo, and A. Gramfort. *Autoreject: Automated artifact rejection for MEG and EEG data, (2017)*. NeuroImage, Volume 159, Pages 417-429, ISSN 1053-8119,

[30] J. Serrà, S. Pascual, and A. Karatzoglou. *Towards a universal neural network encoder for time series (2018)*. CCIA, 120-129.

[31] N. Bigdely-Shamlo, T. Mullen, C. Kothe, K.M. Su, and K.A. Robbins. *The PREP pipeline: standardized preprocessing for large-scale EEG analysis, (2015)*. Frontiers in Neuroinformatics, Volume 9, ISSN 1662-5196.

[32] D. P. Kingma and J. Lei Ba. *Adam: A method for stochastic optimization (2015)*. arXiv:1412.6980v9.

[33] Y. Song, X. Jia, L. Yang, and L. Xie. *Transformer-based Spatial-Temporal Feature Learning for EEG Decoding (2021)*. https://github.com/eeyhsong/EEG-Transformer Accessed Nov. 2022.

[34] D. Truong, M. Milham, S. Makeig, and A. Delorme. *Deep Convolutional Neural Network Applied to Electroencephalography: Raw Data vs Spectral Features (2021)*. https://arxiv.org/abs/2105.04762.

[35] B. Roach and D. Mathalon. *Event-Related EEG Time-Frequency Analysis: An Overview of Measures and An Analysis of Early Gamma Band Phase Locking in Schizophrenia (2008)*. Schizophrenia Bulletin, 34(5), 907-926. doi: 10.1093/schbul/sbn093.

[36] B. Calabrese. *Data Reduction. (2019)*. Encyclopedia Of Bioinformatics and Computational Biology 480-485. doi: 10.1016/b978-0-12-809633-8.20460-3.

[37] D. Malmgren-Hansen, A. Kusk, J. Dall, A. A. Nielsen, R. Engholm, and H. Skriver. *Improving SAR Automatic Target Recognition Models With Transfer Learning From Simulated Data (2017)*. IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 9, pp. 1484-1488, doi: 10.1109/LGRS.2017.2717486.

[38] T. Rojat, R. Puget, D. Filliat, J. Del Ser, R. Gelin, and N. Díaz-Rodríguez. *Explainable Artificial Intelligence (XAI) on TimeSeries Data: A Survey (2021)*. arXiv: 2104.00950.

Appendix

Automatic processing algorithms

The processing algorithms *Autoreject* and *RANSAC* are compared from Figure A1 and A2 respectively. In this example Autoreject rejected more trials and has fewer artifacts remaining, resulting in a cleaner signal.

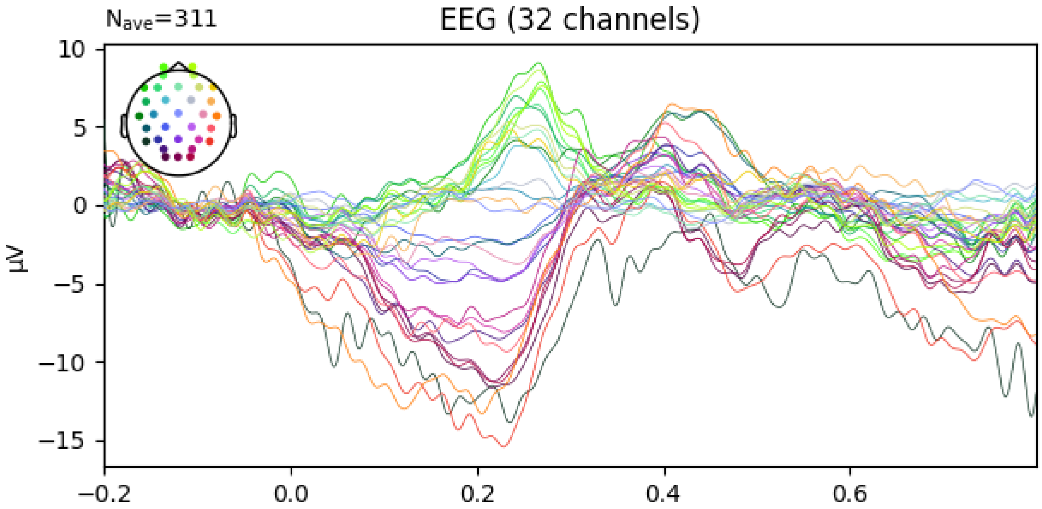

Figure A1: The combined ERP signal of 32 EEG channels. The epochs are processed with the *Autoreject* algorithm.

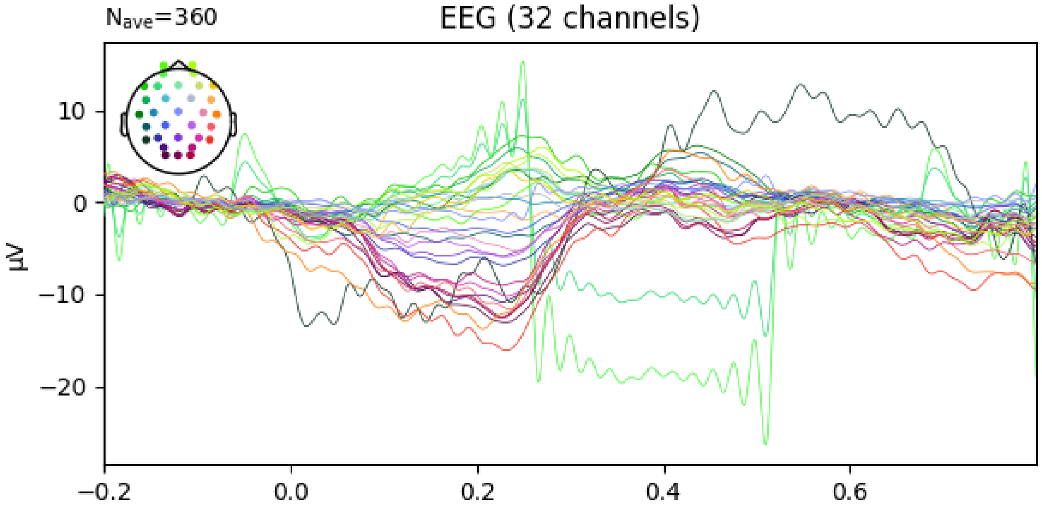

Figure A2: The combined ERP signal of 32 EEG channels. The epochs are processed with the *RANSAC* algorithm.

Valid trials in each dataset

The number of remaining valid trials in each experiment after pre-processing are plotted in a histogram. Figure A3 contains the experiments of the ePodium dataset and Figure A4 contains the experiments of the DDP dataset..

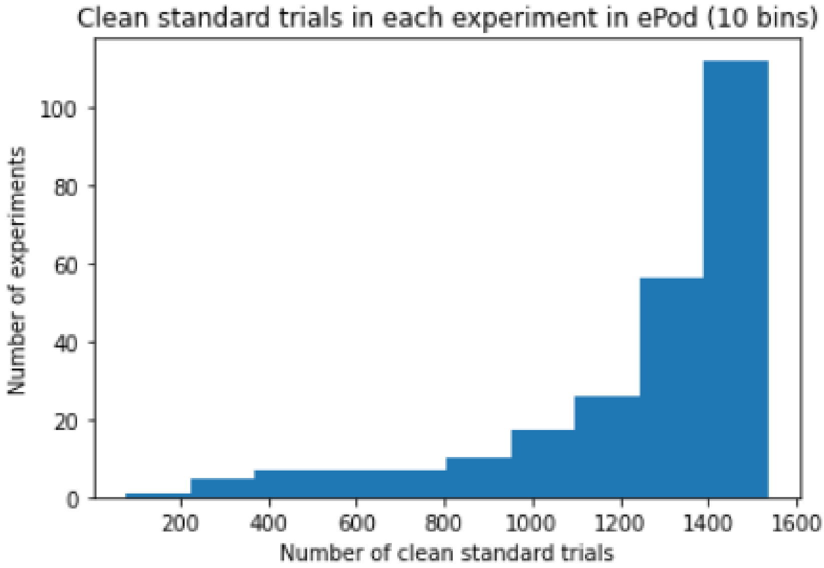

Figure A3: This histogram shows the number of experiments that have a certain amount of *standard* trials in the ePodium dataset after cleaning.

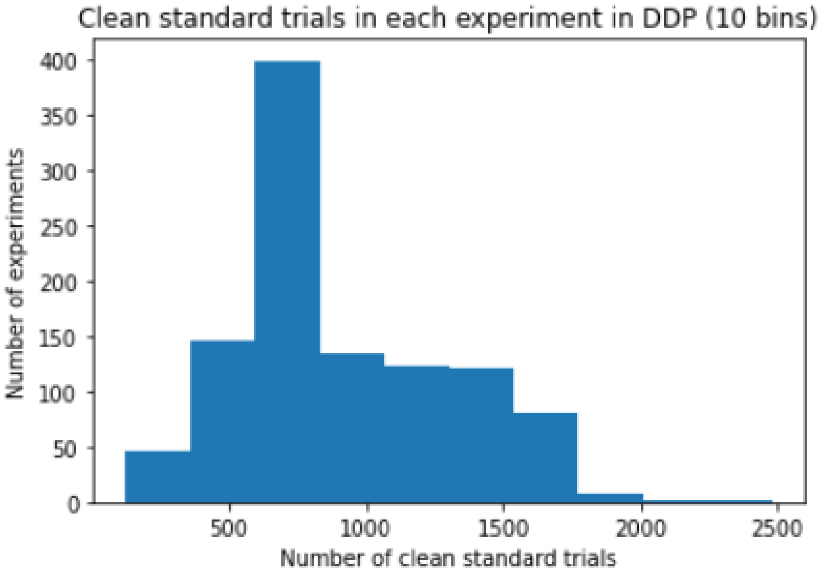

Figure A4: This histogram shows the number of experiments that have a certain amount of *standard* trials in the DDP dataset after cleaning.

Comparing ERP signals with different numbers of trial averages

The ERPs with different number of trial averages in Figure A5 and A6 are from the same participant. From this comparison it is clear that more averages reduces the noise and the magnitude of the voltages. Note that the range in the y-axis is higher in Figure A5 than in Figure A6.

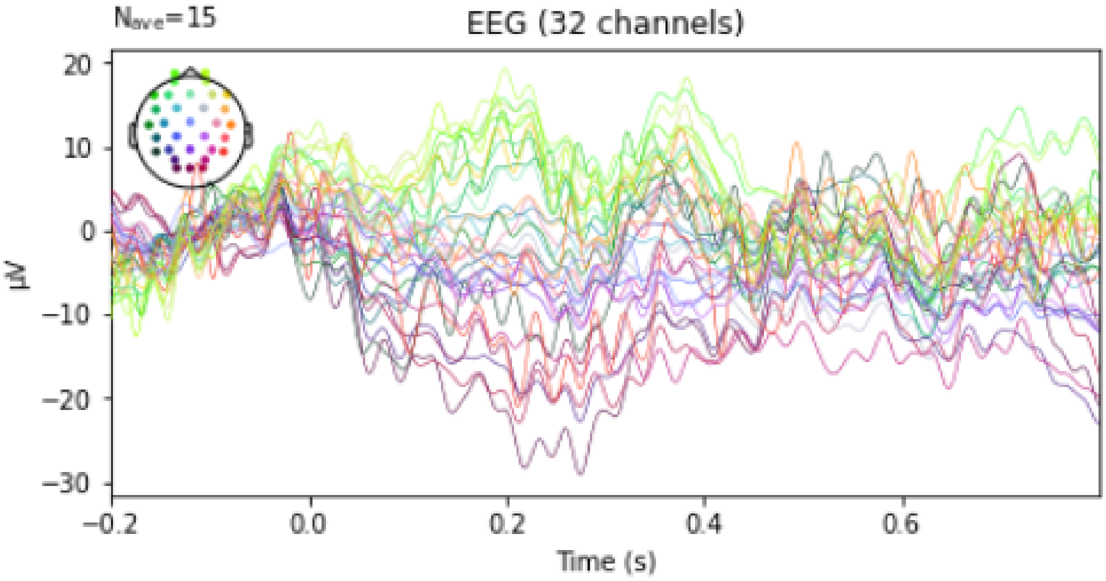

Figure A5: An ERP as the average of 15 trials.

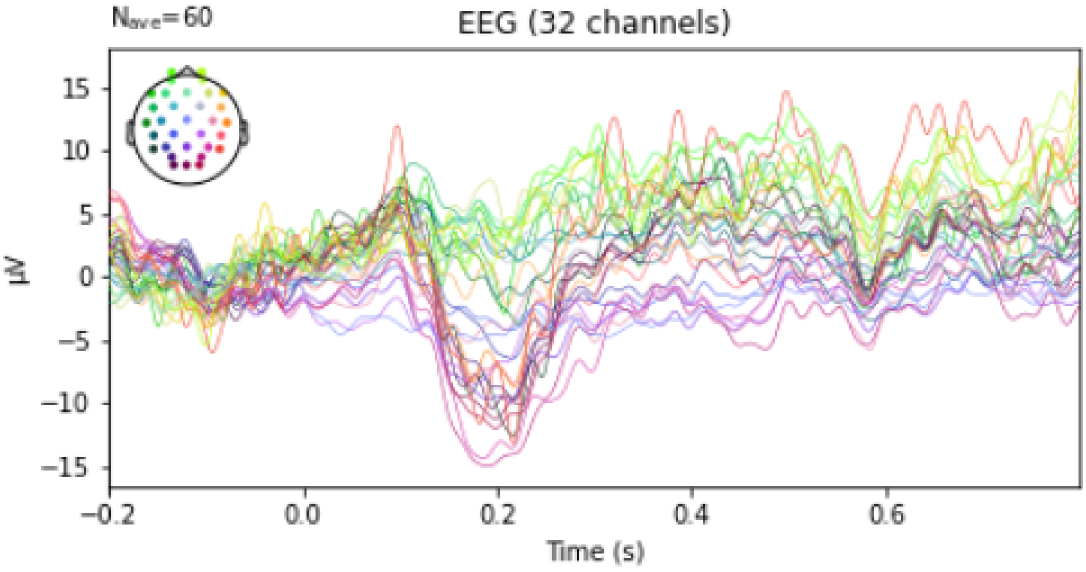

Figure A6: An ERP as the average of 60 trials.

Visualisation of Gaussian artificial noise

In the figures below an ERP channel is plotted. Figure A7 contains no artificial noise, while Gaussian noise with a standard deviation of one μV is added to each time-point in Figure A8.

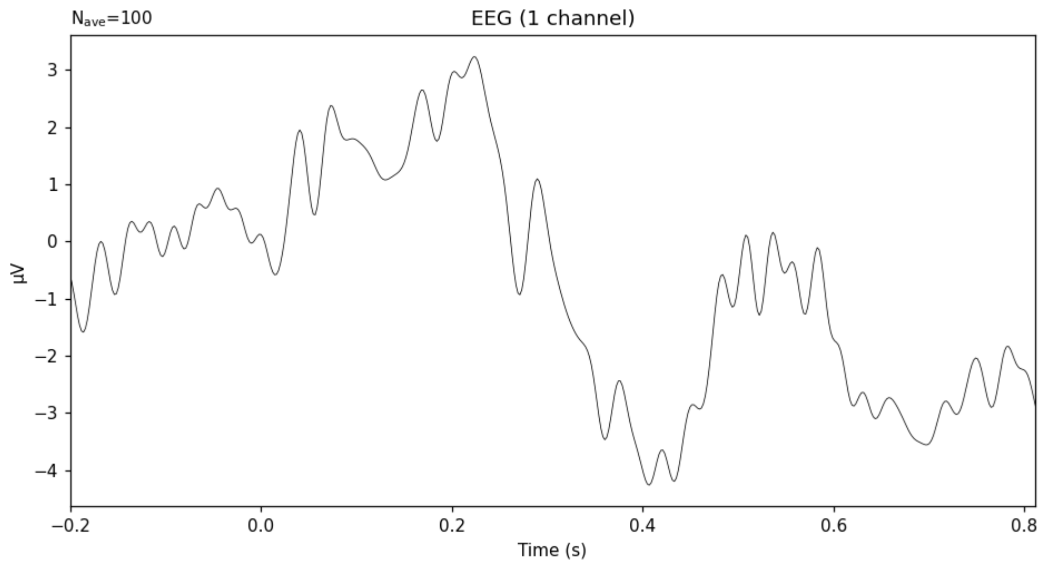

Figure A7: Single ERP channel without noise.

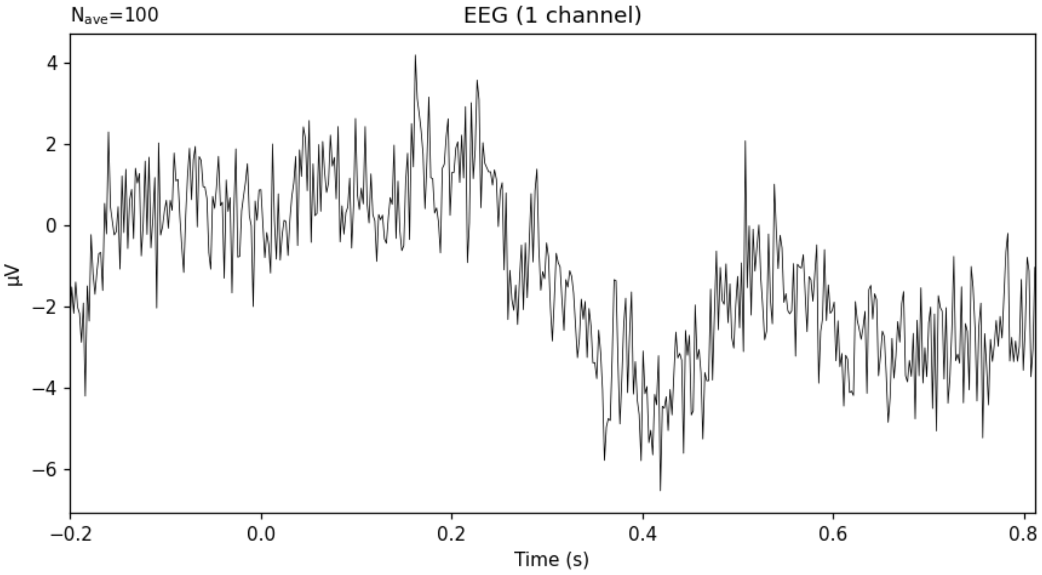

Figure A8: Single ERP channel with 1 μV of Gaussian noise added to each time-point.

Simulated Data

Fourier's theorem states that any continuous function can be expressed as a sum of sines and cosines. The Python notebook 'simulated_data.py' can simulate data that resembles EEG data by adding sines of different frequencies to create the signal. This frequencies can be sampled from any frequency distribution.

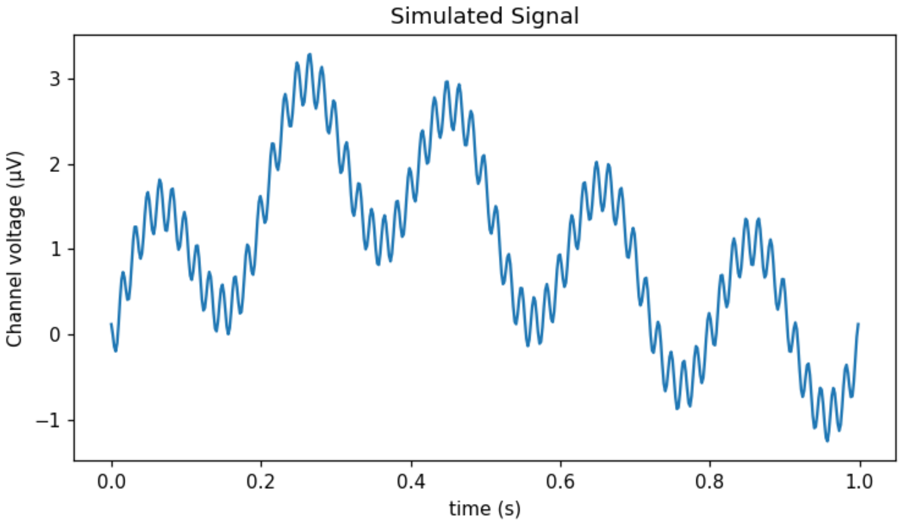

Figure A9: A simulated signal as a sum of multiple sinusoidal waves with both high and low frequencies with different amplitudes.

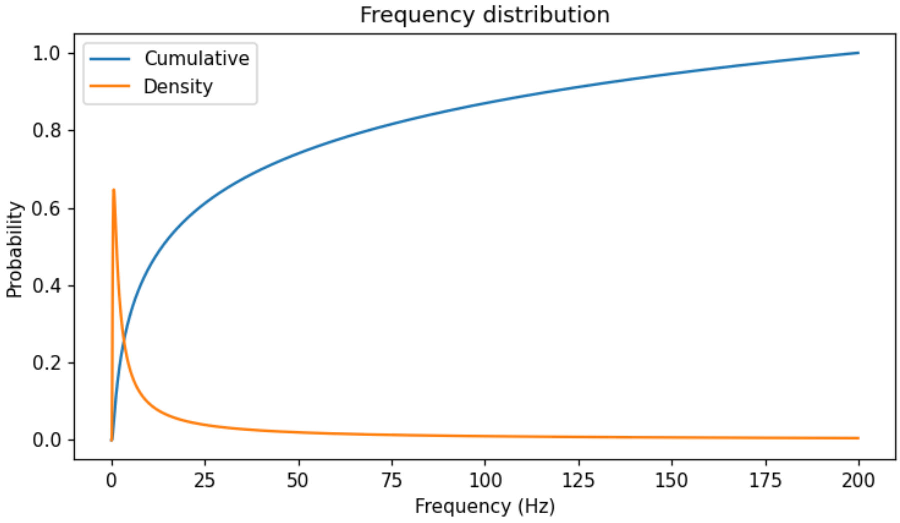

Figure A10: The frequency distribution of the Fourier transform of a simulated one-second EEG signal.

Figure A11 contains a sampled range of frequencies from the frequency distribution in Figure A10. The corresponding signal to this Fourier transform can be found in Figure A12.

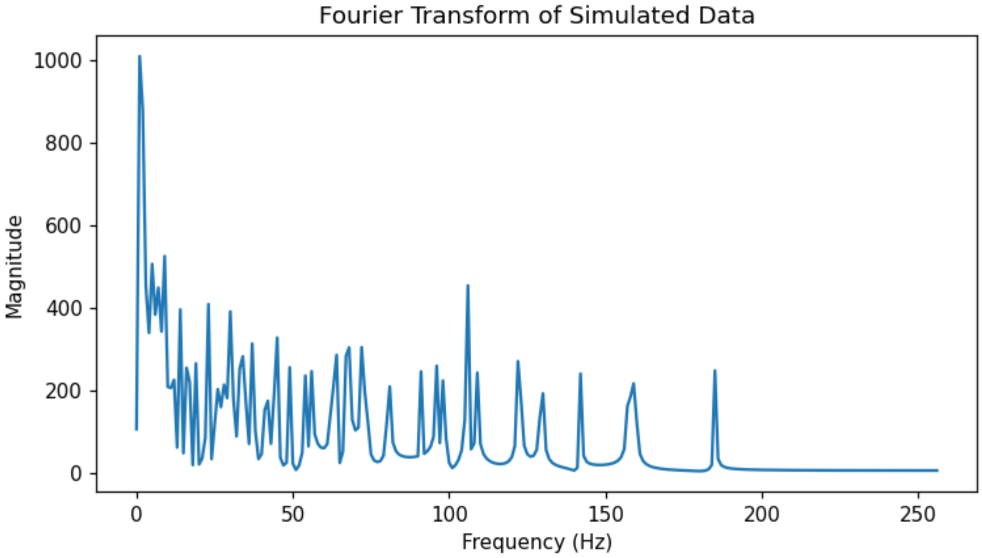

Figure A11: The Fourier transform with the frequencies sampled from the frequency distribution of Figure A10.

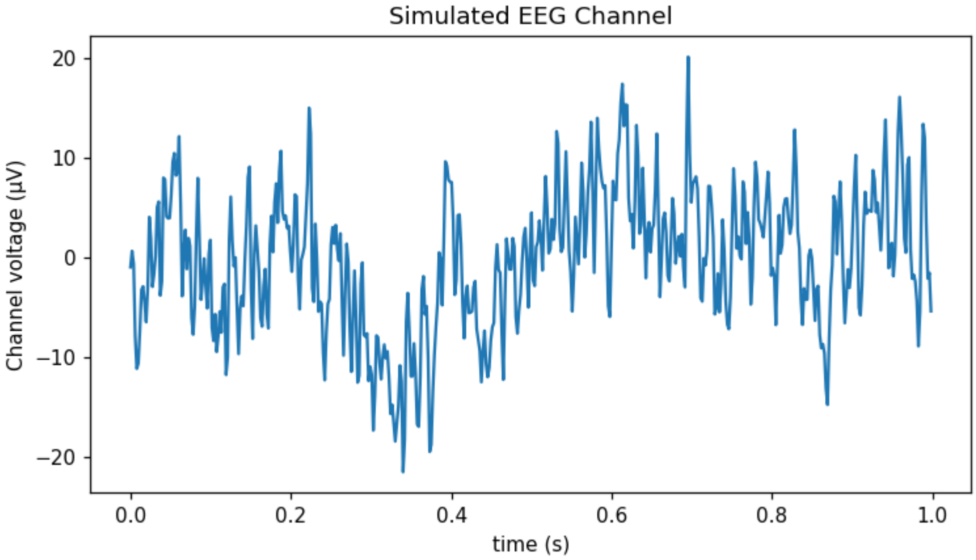

Figure A12: The resulting simulated EEG signal from the Fourier transform of Figure A11.