Evaluation of non-stationary signal processing methods for binary EEG classification

Amanda Ledell

Master's thesis
2022:E78

Lund University

Faculty of Science
Centre for Mathematical Sciences
Mathematical Statistics

CENTRUM SCIENTIARUM MATHEMATICARUM

Populärvetenskaplig sammanfattning

Elektroencefalografi (EEG) är en metod för att mäta hjärnans aktivitet genom att placera elektroder i hårbotten på en individ. Metoden är noninvasiv, relativt billig samt enkel att implementera, och är därför populär inom både forskning och medicin. EEG kan exempelvis användas för tankestyrning av robotproteser, känsloanalys, samt diagnostisering av olika sjukdomstillstånd såsom epilepsi.

För att avgöra om det finns ett samband mellan olika områden i hjärnan vid bearbetning av information, så som ljud, analyserar forskare ofta relationen mellan två elektroder. Om ett sådant sammanband existerar borde de två mätningarna vara fas kopplade, vilket innebär att det finns en konstant tidsfördröjning under tiden information bearbetas. Att identifiera faskopplingar är ett viktigt verktyg i den matematiska analysen av hjärnsignaler som kräver avancerade algoritmer och metoder. En del metoder har idag svårt att separera sanna samband från falska, då det finns en risk att två elektroder mäter samma signal. Detta kan exempelvis ske om elektroderna är placerade nära varandra, och kallas för volume conduction.

Den här uppsatsen jämför och utvärderar olika signalbehandlingsmetoder som är lämpliga vid EEG analyser. Majoriteten av metoderna baseras på fasskillnaden mellan två hjärnsignaler. Flera av metoderna är även framtagna för att hantera volume conduction. Samtliga metoder appliceras först på simulerad data, för att säkerställa deras förmågor, innan de appliceras på två riktiga EEG mätningar. Samtliga metoder undersöks i förhållande till två frågeställningar: 1. Hur bra kan metoden avgöra om en person exponeras för en visuell eller auditiv simulus? 2. Givet att en person får höra ett ord i ett av sina öron, hur väl kan metoden urskilja vilket öra?

Sammanfattningsvis är det lättare att besvara frågeställning ett än två för både simulerad och riktig EEG data. Den bästa metoden för att avgöra närvaron av en stimulus är intressant nog inte baserad på fasskillnaden utan energiskillnaden mellan två signaler. Volume conduction metoderna är de bästa på att avgöra vilket öra som ordet sas i, och klarar även av att identifiera volume conduction. Slutligen, på riktig EEG data klarar den bästa kombinationen av metoder av att avgöra om en stimulus är närvarande i 68.7% av fallen, samt avgöra rätt öra i 55.1% av fallen.

Abstract

Electroencephalogram (EEG) measurements are notoriously noisy and non-stationary and there are several specialized techniques for their analysis and interpretation. In this thesis, we implement a collection of stationary and non-stationary methods including coherence, Phase Locking Value (PLV), Phase Lag Index (PLI), and their imaginary counterparts. In particular, we use the Singular Spectrum Decomposition (SSD) algorithm to decompose each recording into interpretable components before computing a variation of the PLV. All methods are evaluated on simulated EEG data in relation to two research questions; one, how well do they manage to detect whether a subject is presented with a stimulus, and two, given that an auditory stimulus is present in one of the subject's two ears, how well can they determine the side. To measure performance, we train three classification algorithms on features extracted from the above-mentioned methods. We find that the imaginary coherence and imaginary PLV are the best predictors for answering research question two by estimating the sign of the phase difference, whereas the SSD algorithm yields the most important feature for stimulus detection. Lastly, we apply our methods to two sets of real EEG data where it is confirmed that imaginary coherence counteracts volume conduction. In addition, all classification algorithms perform more or less the same but the best one manages to predict the presence of an auditory stimulus with 68.7% accuracy, and the side that the stimulus originated in with 55.1% accuracy.

Acknowledgments

I would first like to express my gratitude to my supervisor Prof. Maria Sandsten for her guidance, time, and helpful insights. Secondly, I would like to express my gratitude to Mikael Johansson, professor of psychology at Lund University, for providing me with interesting EEG data.

Finally, I would like to thank my family and friends for their love and support throughout this thesis and my education. I am especially thankful to Simon Halvdansson and Rebecca Lütz, my time in Lund would not have been the same without them.

Amanda Ledell

List of Abbreviations

ADAM	Adaptive Moment Estimator
ANN	Artificial Neural Network
ciPLV	Corrected Imaginary Phase Locking Value
DFT	Discrete Fourier Transform
EEG	Electroencephalogram
FFT	Fast Fourier Transform
iPLV	Imaginary Phase Locking Value
MLE	Maximum Likelihood Estimation
ML	Machine Learning
NMSE	Normalized Mean Square Error
PLI	Phase Lag Index
PLV	Phase Locking Value
PLV$_{\text{amp}}$	Phase Locking Value with Amplitude
PSD	Power Spectral Density
RF	Random Forest
ROC	Receiver Operating Characteristic
SGD	Stochastic Gradient Descent
SNR	Signal to Noise Ratio
SSA	Singular Spectrum Analysis
SSD	Singular Spectrum Decomposition
SVD	Singular Value Decomposition
wPLI	Weighted Phase Lag Index

Contents

1 Introduction

Understanding the intricacies of the brain has many applications ranging from diagnostics and clinical evaluation to foundational neuroscience and brain-computer interfaces. The first non-invasive method of measuring the brain's activity was pioneered by Angelo Mosso in the late 1800s [30]. Nowadays, the electroencephalogram (EEG) is one of the most popular non-invasive methods of measuring brain activity due to its simplicity, relatively low cost, and high temporal resolution [13, 21, 29].

When neurons in the brain are activated, a local current is generated which the EEG measures via electrodes placed on the scalp of a subject [1]. The constant firing of neurons results in an oscillating signal, consisting of several components known as brain waves. These brain waves are divided into five different groups based on the frequency range of their oscillations and the type of activity, with which they are associated. Table 1 shows a summary of these five groups.

Table 1: Characteristics of the different types of brain waves, from [1, p. 21].

Name	**Frequency**	**Activity**
Gamma	> 35 Hz	Concentration
Beta	$12 - 35$ Hz	Anxiety, active, external attention, relaxed
Alpha	$8 - 12$ Hz	Very relaxed, passive attention
Theta	$4 - 8$ Hz	Deep relaxed, inward focused
Delta	$0.5 - 5$ Hz	Sleep

Recent research has revealed that alpha waves are not only associated with a relaxed state but are actually involved in more cognitive tasks as well. Waves around 10 Hz seem to occur in conjunction with other brain regions and function as an initializer across brain regions [3, p. 227].

As brain currents are inherently time-dependent, we can view brain waves as signals and use signal-processing methods to analyze and interpret EEG measurements. A signal is said to be stationary if the properties of the signal, such as its frequency or mean, are time-invariant and non-stationary otherwise. Many methods used in EEG analysis are built on the assumption of stationarity which in reality is rarely the case, as can be seen in Figure 1 where the mean of the EEG measurement clearly varies over time. Using a stationary method on non-stationary data can certainly yield interesting results but the results may not be valid. This is due to the fact that violating the assumptions of a statistical method can influence the accuracy and reliability of the results [17]. In order to justify the validity of EEG analysis, developing more user-friendly non-stationary methods is therefore crucial.

Spectral coherence is a stationary method that is frequently used within neuroscience to determine if two brain regions are interacting when for example processing sound [32]. An alternative method to spectral coherence is the phase locking value approach, which is better suited to neural activity since it does not require any assumptions on the data [23]. This method is the most popular non-stationary, phase-based method in EEG analysis but it has its own drawbacks [7].

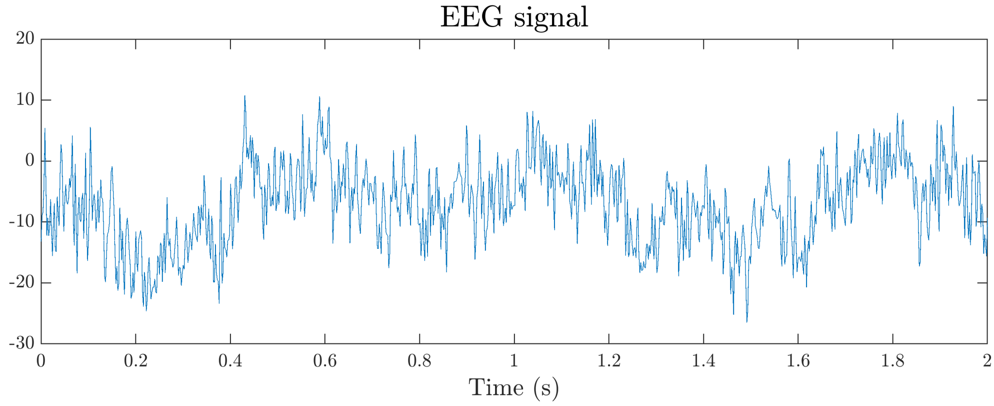


Figure 1: An example of a two-second long EEG signal.

Aim of thesis

The aim of this thesis is to determine suitable and reliable methods for EEG analysis. A total of eleven signal processing methods will be evaluated, nine out of which are non-stationary. The two stationary methods are variations on spectral coherence while the remaining nine are based on phase-coupling. Furthermore, a total of 18 features will be extracted from the methods and used in three classification algorithms.

All methods are evaluated against two research questions: One, how well do the methods manage to determine if a person is presented with a stimulus. Two, for an auditory stimulus, how well can the methods determine from which side the stimulus originated. In order to achieve the intended aim of the thesis we first describe the different methods in Section 2 before evaluating them on a simulated data set in Section 3. Finally, we apply the methods to two real EEG data sets in Section 4 to see how well the methods perform.

2 Theory

2.1 Stationary Signal Processing

Digital signal processing is a relatively new field which became popular in 1965 when the Fast Fourier Transform (FFT)-algorithm made it possible to efficiently compute the Discrete Fourier Transform (DFT) on computers [10]. Given an input signal $x(t)$ and n data points, we define the DFT as

$$\tilde{x}(f) = \sum_{t=0}^{n-1} x(t)e^{-i2\pi ft}, \tag{2.1}$$

which enables us to analyze a signal in the frequency domain. This tends to be done by computing the (Power) Spectral Density (PSD), also referred to as an auto spectrum, which is the DFT of the covariance function. The auto spectrum shows the distribution of different frequencies and is most commonly estimated by the periodogram [22, pp. 83, 238], computed as

$$\Phi_{xx}(f) = \frac{1}{n}|\tilde{x}(f)|^2, \tag{2.2}$$

i.e., the normalized square of the DFT.

2.1.1 Spectral Coherence

Spectral coherence is used to detect correlations between two signals, for example, two areas of the brain in neuroscience [23]. Let $x(t)$ and $y(t)$ be two stationary signals with zero-mean, and $\tilde{x}(f)$ and $\tilde{y}(f)$ be their respective Fourier transform. We can rewrite Equation (2.2) to get their combined cross-spectrum as

$$\Phi_{xy}(f) = \frac{1}{n}\tilde{x}(f)\overline{\tilde{y}(f)}, \tag{2.3}$$

where $\overline{\tilde{y}(f)}$ is the complex conjugate of $\tilde{y}(f)$ [24]. Equation (2.3) can be decomposed as

$$\Phi_{xy}(f) = A_{xy}(f)e^{i\theta_{xy}(f)}, \tag{2.4}$$

where A_{xy} is the amplitude spectrum and θ_{xy} is the phase spectrum. The complex coherence spectrum is then defined as,

$$\gamma_{xy}(f) = \frac{\Phi_{xy}(f)}{\sqrt{\Phi_{xx}(f)\Phi_{yy}(f)}}. \tag{2.5}$$

The coherence spectrum, also known as the magnitude coherence spectrum or coherence [8], is simply the magnitude of $\gamma_{xy}(f)$,

$$\mathcal{C}_{xy}(f) = |\gamma_{xy}(f)| = \frac{|\Phi_{xy}(f)|}{\sqrt{\Phi_{xx}(f)\Phi_{yy}(f)}}, \qquad 0 \leq \mathcal{C}_{xy}(f) \leq 1. \tag{2.6}$$

Algorithm 1 Coherence spectrum

```
1. For each window $k$ repeat the following steps.
    (a) Compute the FFT of both $x$ and $y$ multiplied with the $k^{\text{th}}$ window.
    (b) Compute and store the individual auto spectrum of $\tilde{x}$ and $\tilde{y}$.
    (c) Compute and store the cross-spectrum between this $\tilde{x}$ and $\tilde{y}$.
2. Sum the $K$ cross-spectrums and auto spectrums, $\sum_{k=1}^{K} \Phi_{xy}(f)^{(k)}$, $\sum_{k=1}^{K} \Phi_{xx}(f)^{(k)}$ and $\sum_{k=1}^{K} \Phi_{yy}(f)^{(k)}$.
3. Compute the coherence spectrum $\mathcal{C}_{xy}(f)$ according to Equation (2.6).
```

It is possible to use $\mathcal{C}_{xy}(f)$ as a metric of how strong the linear dependency is between $x(t)$ and $y(t)$, where a value close to 1 indicates a strong dependency. There are some variations in the literature on whether coherence is defined as in Equation (2.6) or whether it should be the squared magnitude [15, 24]. Both definitions are widely used, bounded, and can be used as a statistical tool for measuring correlation or dependency. This thesis uses the convention of Equation (2.6).

A drawback with the periodogram, and thus $\mathcal{C}_{xy}(f)$, is its large bias and variance, leading to bad spectral density estimations. The modified periodogram therefore includes a window function w when computing the DFT of the signal, which reduces the bias compared to the normal periodogram [8]. There are different kinds of windows but the most common one is the Hanning window, which is smooth [22, p. 248]. Furthermore, $\mathcal{C}_{xy}(f)$ has a constant value of one when computed using the periodogram or the modified periodogram, which does not yield any meaningful results.

A variance reduction can be achieved by segmenting the data into K overlapping segments and then averaging their respective auto spectrum in order to receive a new averaged estimate, $\widehat{\Phi_{xx}}(f)$. The Welch method combines these two approaches [37] and can be written as

$$\widehat{\Phi_{xx}}(f) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{n} \left| \sum_{t=0}^{n-1} x(t) w_k(t) e^{-i2\pi f t} \right|^2 , \tag{2.7}$$

where $x(t)$ is the signal and $w_k(t)$ is the time-shifted window function. Computing $\mathcal{C}_{xy}(f)$ based on the Welch method does not yield a constant value of one for all frequencies due to its averaging properties [22, p. 256]. It is common to have overlapping segments with a 50% overlap. Such overlap has shown significant improvements in terms of both bias and variance reductions [9]. Increasing the overlap also increases the computational costs since we need to perform more FFTs. However, the cost associated with 50% overlaps is deemed reasonable in relation to the bias and variance improvements [8]. In addition, this method does not cause $\mathcal{C}_{xy}(f)$ to always equal one. The approach to estimating the coherence spectrum is detailed Algorithm 1.

2.1.2 Imaginary Coherence

Coherence measures the frequency distribution and correlation between two signals. The method is however not always trustworthy, as false connectivity may arise, resulting in wrongful conclusions. This can occur in EEG analysis if electrodes measure the same underlying source and it is interpreted as true connectivity between two brain regions. The problem is known as *volume conduction* and tends to occur when electrodes are placed close to each other [33].

In 2004, *imaginary coherence* was introduced in [24] as a possible solution and is defined as the imaginary part of the complex coherence,

$$i\mathcal{C}_{xy}(f) = \mathrm{Im}\left\{\gamma_{xy}(f)\right\} = \frac{\mathrm{Im}\{\Phi_{xy}(f)\}}{\sqrt{\Phi_{xx}(f)\Phi_{yy}(f)}}. \tag{2.8}$$

The reasoning for this method goes as follows: When volume conduction occurs, we only have one true source that two electrodes end up measuring. A signal cannot be time-lagged to itself and we should therefore not expect any time lag in such a scenario. Moreover, no time lag means that the relative phase is either 0 or $\pm\pi$, depending on the sign of the electric potential and the electrodes. Hence, the coherence $\mathcal{C}_{xy}(f)$ spectrum is in such a case real-valued, which in turn implies that its imaginary component cannot detect signals from common-sources [24, 38].

While imaginary coherence can resolve the problem of false connectivity, it cannot measure the strength of correlations as well as coherence [33]. Additionally, imaginary coherence may also fail to recognize connectivity when time-lags are very small or vanishing, but it is believed to be a justified risk due to its insensitivity to same source measurements [24]. Combining the two forms of coherence may therefore yield a better understanding of the connectivity between two signals $x(t)$ and $y(t)$ since $\mathcal{C}_{xy}(f)$ indicates the strength of their coupling whereas $i\mathcal{C}_{xy}(f)$ determines if the connectivity resembles true brain interaction.

2.2 Non-stationary Signal Processing

As previously mentioned, coherence methods assume stationarity which is rarely the case for EEG signals. They are known to be noisy and non-stationary but the stationarity assumption is frequently applied in neuroscience studies which can lead to unreliable estimates [23]. We therefore introduce another non-stationary approach that instead looks at the phase correlation between two signals and does not make any assumptions about the data. This approach was motivated by previous research, which showed that different areas of the brain enter a time period of phase-locking when interacting with a particular stimulus [20]. In other words, the idea is to investigate whether there is a change in the amount of phase synchronization in the brain during the time that a stimulus is present.

2.2.1 Phase Locking value

The *Phase Locking Value* (PLV) was the first phase-based method introduced in [20] in 1999 and can be computed across trials and/or over time. It is most commonly

Algorithm 2 Phase Locking Value

```
Compute the following for all $N$ trials simultaneously.
    1. Apply a band-pass filter to both signals to get $\tilde{x}$ and $\tilde{y}$ and let the filter be centered between ±2 Hz of the desired frequency.
    2. Compute the respective Hilbert transforms of both filtered signals,
           $\hat{x} = \tilde{x} + i \cdot \text{Hilbert}(\tilde{x}), \qquad \hat{y} = \tilde{y} + i \cdot \text{Hilbert}(\tilde{y}).$
    3. Compute the phases of both Hilbert transforms to get $\theta_x$ and $\theta_y$.
    4. Compute the PLV as in Equation (2.9).
```

computed across trials since two brain areas are believed to have underlying connectivity if they, for a given time period, show a consistent phase difference [20]. We compute PLV across trials as

$$\text{PLV}_{x,y}(t) = \frac{1}{N}\left|\sum_{n=1}^{N} e^{-i(\theta_x(t,n)-\theta_y(t,n))}\right|, \qquad 0 \leq \text{PLV}_{x,y}(t) \leq 1, \tag{2.9}$$

where we have N trials and $\theta_x(t,n)$ is the phase of the signal x at time t for the n^{th} trial. This results in PLV as a function of time, which can be useful in for example epilepsy studies but not in machine learning applications. It is however possible to compute PLV across time which yields N different scalar PLVs. Before the computations in Equation (2.9) are carried out, we need to design and apply a narrow band-pass filter since any form of meaningful results is only possible when we have well-defined oscillations [18]. The oscillations must also be within the frequency range of interest, and filtering is thus often considered the greatest challenge with this method [23]. The whole process is summarized in Algorithm 2.

A higher PLV indicates a stronger phase synchronization, but it can not be used as a direct measurement of the correlation between signals as coherence can. For instance, say that we measure the PLV between the channel pair (i, j) and (k, l) during the time that the stimulus is present and obtain $\text{PLV}_{i,j} = 0.7$ and $\text{PLV}_{k,l} = 0.8$. These results do not necessarily mean that channels k and l have a greater significant phase synchronization during the time period of interest. It could be that these two channels simply have a continuously strong synchronization due to volume conduction which is not affected by the stimulus. It is therefore common to perform a hypothesis test to see if there is a significant change in PLV during the time that the stimulus is present, compared to before. Depending on the purpose of the analysis, some tests may be more suitable than others. The original paper suggested the Raleigh test [20] whereas other papers use the z-test [18]. However, PLV on its own is still a good indicator of how the oscillations in the brain are relating to one another, but it is hard to determine what results should be deemed significant.

2.2.2 Imaginary PLV

The risks associated with volume conduction are also relevant when estimating PLV. The logic applied to imaginary coherence in Section 2.1.2 to remove zero-lag effects from normal coherence, can also be applied to PLV [26]. This line of reasoning is justified by the strong similarities between PLV and coherence, and a thorough explanation can be found in [7]. The *imaginary* PLV is therefore

$$\mathrm{iPLV}_{x,y}(t) = \frac{1}{N}\mathrm{Im}\left\{\sum_{n=1}^{N} e^{-i(\theta_x(t,n)-\theta_y(t,n))}\right\}. \tag{2.10}$$

The iPLV is not normalized which can become problematic in cases of volume conduction [7]. The *corrected imaginary* PLV (ciPLV) is therefore introduced as an alternative to iPLV,

$$\mathrm{ciPLV}_{x,y}(t) = \frac{\frac{1}{N}\mathrm{Im}\left\{\sum_{n=1}^{N} e^{-i(\theta_x(t,n)-\theta_y(t,n))}\right\}}{\sqrt{1-\left(\frac{1}{N}\mathrm{Re}\left\{\sum_{n=1}^{N} e^{-i(\theta_x(t,n)-\theta_y(t,n))}\right\}\right)^2}}. \tag{2.11}$$

Additionally, ciPLV has been shown to outperform iPLV in simulations where volume conduction or source leakage is present [7], indicating that ciPLV is a good metric for EEG analysis.

2.2.3 Phase-Amplitude PLV

As previously described, PLV can be computed between two brain areas in order to determine if they show any interaction based on their coupling strength. There is however instances of brain cross-frequency coupling where the same brain region experiences interactions between different brain waves, for example, theta and gamma [35]. In order to assess and evaluate these forms of coupling, another form of the PLV is needed, which instead looks at a single signal $x(t)$ filtered at two different frequency bands. The signal is first filtered with a narrow low-frequency band and then again with a wider high-frequency band, since, the amplitude of the wider filtered signal should oscillate at the narrow filtered signal's frequency if cross-frequency coupling exists [18].

We refer to this metric as $\mathrm{PLV}_{\mathrm{amp}}$, and write it formally as

$$\mathrm{PLV}_{\mathrm{amp}} = \frac{1}{T}\left|\sum_{t=1}^{T} e^{-i\left(\theta_{\mathrm{low}}(t)-\theta_{\mathrm{high}}(t)\right)}\right|, \tag{2.12}$$

where T is the total number of time points. In contrast to PLV, $\mathrm{PLV}_{\mathrm{amp}}$ is computed over time and the θs are estimated differently. The phase from the higher frequency is obtained by computing the amplitude of the analytic signal and then performing a second Hilbert transform after which the phase θ_{high} is extracted. The phase extracted from the low-frequency range, θ_{low}, is computed in the same manner as described in Algorithm 2 [18]. The full procedure is detailed in Algorithm 3.

Algorithm 3 Amplitude focused Phase Locking Value

1. Apply a narrow low-frequency band-pass filter to signal, x to get $\hat{x}_{\text{low}}$. Let the filter be of width 2 Hz and centered at the desired frequency.

2. Compute the Hilbert transform of $\hat{x}_{\text{low}}$ and extract the phase as θ_{low}.

3. Apply a wide high-frequency band-pass filter to signal x, to get $\hat{x}_{\text{high}}$. Let the filter be approximately 20 Hz wide and in the gamma range.

4. Compute the Hilbert transform of $\hat{x}_{\text{high}}$ and extract the phase as θ_{high}.

5. Compute the PLV_{amp} metric as described in Equation (2.12).

2.2.4 Singular Spectrum Decomposition

As previously mentioned in Section 2.2.1, the greatest challenge of the PLV algorithm is to successfully filter and decompose the signal into well-defined components, in order to obtain any meaningful results from the PLV algorithm. An alternative approach to filtering the signal before computing the PLV is to use the Singular Spectrum Decomposition (SSD) algorithm to “automatically” decompose the signal [23]. The SSD algorithm is a relatively new algorithm used to decompose noisy and non-stationary time series and was originally proposed by Bonizzi et al. in 2014 [5]. This thesis is not going to describe the algorithm in detail but a thorough explanation can be found in the original paper [5].

The algorithm is a further development of the Singular Spectrum Analysis (SSA) algorithm and is an adaptive decomposition method. SSA is in turn a principle component analysis (PCA) based method, which means that it looks for directions in data sets with large variances [25], and can be divided into four main steps. Let us first create a so-called trajectory matrix, which is a matrix with constant cross-diagonals, of the original signal $x(t)$ by embedding it into a vector space of dimension M, where $1 < M < N$ and N is the length of $x(t)$. The embedding procedure generates $K = N - M + 1$ lagged vectors of the form

$$\boldsymbol{x}_i = (x(i), \ldots, x(i + M - 1))^T, \quad \text{where} \quad i = 1, \ldots, N - M + 1,$$

which make up the $(M \times K)$-dimensional matrix $\boldsymbol{X}$. Secondly, we apply Singular Value Decomposition (SVD) to the trajectory matrix and thereafter group the principle components based on specific components, such as trend and oscillations, etc., of the original signal. Finally, the SSA algorithm reconstructs the components of the original signal.

The main critique of SSA is that the embedding dimension M and the principal components used for reconstruction must be chosen manually, which is difficult since we do not have any prior knowledge of the data. Both these aspects have however been automated in the SSD algorithm by instead extracting each component until a certain threshold is reached. The embedding dimension M is re-defined at each

Algorithm 4 Singular Spectrum Decomposition

Repeat the following procedure until NMSE $\leq th$, yielding K components. Let j be the iteration number and let v_1 be the original signal.

1. Compute the PSD of the residual time series $v_j(t)$ and determine which frequency produces the greatest peak.

2. Define the embedding dimension, $M_j = \lfloor 1.2 \frac{F_s}{f_{max}} \rfloor$ and generate the trajectory matrix $\boldsymbol{X}$. **Note!** If $j = 1$ and the normalized frequency $\frac{f_{max}}{F_s} < 0.001$, then a sizable trend is deemed to be detected and M is instead set to $M = \lfloor \frac{N}{3} \rfloor$.

3. Compute the SVD of $\boldsymbol{X}$.

4. Identify which subset I_j of components from the decomposition should be used to reconstruct the j^{th} component $g_j(t)$.

    - The frequency band used to identify good principle components is defined as $[f_{max} - \delta f; f_{max} + \delta f]$, where δf is the half-width of f_{max} and unknown. Estimating δf is done by optimization but does in short make use of three Gaussian functions which help to describe the properties of the j^{th} PSD. They account for the dominant peak, the second dominant peak as well as the average of the two.

5. Reconstruct the component series by diagonally averaging $\boldsymbol{X}_{I_j}$ along the cross-diagonals, yielding $g_j(t)$. **Note!** If a sizable trend has been detected, the reconstruction only makes use of the first left and right eigenvectors before the diagonal averaging.

6. If no trend has been detected, repeat steps $2 - 5$ once in order to improve $g_j(t)$ and avoid the risk of $v^{(j+1)}(t)$ having more energy than $v^{(j)}(t)$.

7. Obtain the new residual time series as $v_{j+1}(t) = v_j(t) - g_j(t)$

8. Compute the NMSE for this iteration,

$$\text{NMSE}(j) = \frac{\sum_{n=1}^{N} v^{(j+1)}(n)^2}{\sum_{n=1}^{N} x(n)^2}.$$

   and check if it is greater than th.

iteration and the principle components are selected based on their frequencies being in a defined frequency band. SSD also constructs the trajectory matrix differently. In the new approach, we change the dimension of $\boldsymbol{X}$ to $(M \times N)$, and define each $\boldsymbol{x}_i$ as

$$\boldsymbol{x}_i = (x(i), \dots, x(N), x(1), \dots, x(i-1)), \qquad \text{where} \qquad i = 1, \dots, M.$$

The new formulation of the trajectory matrix creates additional benefits in regards to decreasing the energy of the residuals by enhancing the signals' oscillatory content [5].

The algorithm is repeated until the normalized mean square error (NMSE) between the j^{th} component residual and the original signal is less than a given threshold, th, yielding K components. The default value of th is 0.01 (1%) but can be changed manually. An outline of SSD can be found in Algorithm 4.

2.2.5 Phase Lag Index

Another method used when dealing with volume conduction is the *Phase Lag Index* (PLI) which has strong similarities to both iPLV and $i\mathcal{C}_{xy}(f)$. There are two equivalent definitions for PLI, one resembling PLV and one coherence. PLI is bounded between 0 and 1, where a value of 0 indicates no phase coupling between the signals. The original definition of PLI from [33] is

$$\text{PLI} = \left|\mathbb{E}\left[\text{sgn}\left(\Delta\theta_t\right)\right]\right|, \qquad 0 \leq \text{PLI} \leq 1, \tag{2.13}$$

where $\Delta\theta_t$ is the phase difference between two signals. This resembles PLV due to them both being based on phases. However, PLI is a function of both the variance and mean of $\Delta\theta_t$ whereas PLV is a function of the mean [2]. Previous work also indicates that PLI might identify stronger synchronizations between channels more accurately than iPLV due to a stronger robustness to volume conduction [33]. It is nonetheless beneficial to include both since they are functions of different parameters.

The other definition of PLI, with similarities to imaginary coherence, from [36] is written as

$$\text{PLI} = \left|\mathbb{E}\left[\text{sgn}\left(\text{Im}\left\{\Phi_{xy}(f)\right\}\right)\right]\right|, \tag{2.14}$$

where $\Phi_{xy}(f)$ is the cross-spectrum defined in Equation (2.3). PLI is solely based on the imaginary component of $\Phi_{xy}(f)$, whereas $i\mathcal{C}_{xy}(f)$ is normalized by the amplitude of the signal and thus includes some real components as well [36]. This distinct difference indicates that PLI might be a more suitable metric in terms of dealing with volume conduction but it is however encouraged to include both when possible.

It is also possible to obtain the *signed* PLI (sPLI) by not taking the absolute value in either Equation (2.13) or (2.14) [33, 36]. This may be useful in analyses where we are interested in the order that two signals appear. Meanwhile, other applications may find it desirable to determine if the PLI obtained is significantly greater than zero. This can be done by introducing a surrogate data set which resembles the original data, but lacks any correlation between channels, and then comparing the two with a z-test [33].

One limitation of the PLI is that a small disturbance can turn phase lags into false leads, causing a discontinuity that affects the sensitivity of the measurement [36]. This is addressed by introducing the *weighted* PLI (wPLI) which in addition to PLI is weighted by the imaginary component of the cross-spectrum [26]. We therefore define the wPLI as

$$\text{wPLI} = \frac{\left|\mathbb{E}\left[\text{Im}\left\{\Phi_{xy}(f)\right\}\right]\right|}{\mathbb{E}\left[\left|\text{Im}\{\Phi_{xy}(f)\}\right|\right]} = \frac{\left|\mathbb{E}\left[\left|\text{Im}\left\{\Phi_{xy}(f)\right\}\right|\text{sgn}\left(\text{Im}\{\Phi_{xy}(f)\}\right)\right]\right|}{\mathbb{E}\left[\left|\text{Im}\{\Phi_{xy}(f)\}\right|\right]}. \tag{2.15}$$

The wPLI is less affected by uncorrelated noise sources than PLI and is also able to detect phase-based synchronization changes with a higher accuracy [36].

2.3 Machine Learning

Machine learning (ML) is becoming an increasingly popular branch of mathematics and computer science. There are many facets to ML, one of which is statistical learning where we want to create a model which can predict future values, so called outcomes, based on unseen data. A model is only able to predict outcomes based on the same underlying distribution as the data it was trained on. The general procedure when designing a model is to split the data into a training and test set, then to train the model on the training set before evaluating how well it manages to predict the outcomes of the test set.

There are many different kinds of models and depending on the problem and the data set at hand, some may be more suitable than others. Regression models are for example used on continuous data sets whereas classification models are used on categorical data sets. A model is said to be obtained through *supervised* learning if we have labeled the targets during training and *unsupervised* learning if not. This thesis is only focusing on binary supervised classification models.

Each model consists of different parameters, where some are fixed and set by the programmer whereas others are optimized during training. Optimizing these parameters is actually what is meant by “training” a model. The fixed parameters are called hyperparameters and can greatly impact the optimization procedure, and thus the performance of the model. Choosing suitable hyperparameters is therefore important and this is discussed in Section 2.3.4.

Designing a model that performs well on the training set is relatively easy, whereas it can be significantly harder to create a model that generalizes to unseen data. If a model performs significantly better on the training data compared to the test set, we say that the model is overfitted. This can for example happen if we have too many parameters, or train the model to account for every outlier. There are various strategies available to avoid overfitting and they are commonly referred to as regularization [14, p. 224]. Avoiding overfitting is crucial since an overfitted model is unable to predict outcomes on unseen data.

2.3.1 Logistic Regression

The simplest model used for binary classification problems is the Logistic Regression model and it is based on the probability that an outcome either belongs to class 1 or 0,

$$p(X) = \Pr(Y = 1|X) = 1 - \Pr(Y = 0|X).$$

The probability $p(X)$ is written as

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}, \tag{2.16}$$

where β_0 and β_1 are the parameters we wish to optimize. We can rewrite Equation (2.16), to make it linear in X as

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X, \tag{2.17}$$

where the left hand side is referred to as the *log-odds* [19, p. 132]. The Maximum Likelihood Estimation (MLE) method is used to estimate the parameters by finding values β_0 and β_1 that maximize the probability of making the correct classifications. This is obtained by maximizing the log-likelihood function,

$$\begin{aligned} l(\beta) &= \sum_{i=1}^{N}\left[y_i \log p(x_i;\beta) + (1-y_i)\log(1-p(x_i;\beta)\right] \qquad (2.18)\\ &= \sum_{i=1}^{N}\left[y_i\beta x_i - \log(1+e^{\beta x_i})\right], \end{aligned}$$

with $\beta = \beta_0, \beta_1$ [16, p. 120]. Note that these equations are valid for binary classification, and in cases where Y takes on more than two values multiple logistic regression is needed.

2.3.2 Random Forest

The Random Forest (RF) model is one of the most commonly used classification algorithms today [11]. A RF model consists of B decision trees where each tree is obtained through boosting. This means that each tree is trained on a subset of the original training data, $\mathbf{Z}$, and the final outcome, for classification, is the most popular class among these trees [6]. Finding the outcome in this manner is referred to as *the majority vote* and it helps to reduce the variance of the final model which is desirable [19, p. 316].

As can be seen in Figure 2, each tree consists of several binary splits where each

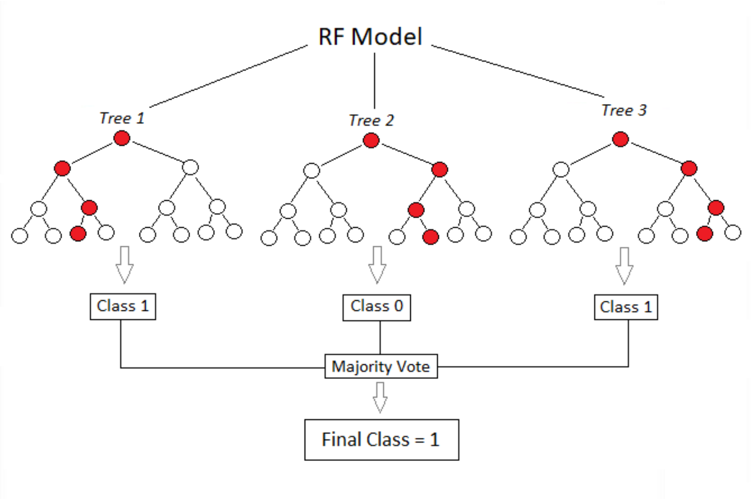


Figure 2: An illustration of a random forest with three trees and three splits.

Algorithm 5 Random Forest

```
1. Create $B$ number of trees by repeating the following steps.
    (a) Draw bootstrap sample to create a new data set $\mathbf{Z}$ of length $N$.
    (b) Grow a tree on the data set $\mathbf{Z}$ by repeating the following steps for each terminal node until either the minimum node size, $t$, or the maximum number of splits $S$ is reached.
        i. Randomly select $\sqrt{N}$ features to use when considering the split of this node.
        ii. Pick the best feature and split the node into two new nodes.
2. Predict the class by taking the majority vote of the B trees.
```

split represents two cases of one feature. For example, say that we have the feature age, then a possible distinction could be whether or not age is greater than 50. Determining which feature to use and what the distinction should be is done by minimizing a loss function, the most common of which is the *Gini Index.* Each split considers m randomly selected features which help to decorrelate the trees and yield a more reliable final model [19, p. 319]. A common choice is to have $m = \sqrt{N}$ where N is the total number of features [16, p. 592].

It is possible to create a forest without predefining a lot of hyperparameters, apart from the number of trees B, but it is advisable to include some. Deciding on a maximum number of splits S, or a minimum node size t, can for example help to regularize the model. A general description of the RF model is available in Algorithm 5.

One advantage of the RF model is that it provides an easy overview of the importance of different features. The feature with the largest decrease in its loss function after a split is considered to be the most important [19, p. 319]. The importance of features is often viewed graphically in a so-called *feature importance* plot.

2.3.3 Artificial Neural Network

Artificial neural networks (ANN) are in general more advanced than the models previously presented, and tend to be applied to larger data sets due to their complexity. This model, or network rather, can be thought of as a spider web with an input layer, an output layer, and H hidden layers, where H is a hyperparameter, and increasing it yields a more complex model. These kinds of networks are inspired by the structure of the human brain where the nodes, denoted by the circles in Figure 3, correspond to neurons, hence the name *neural* networks.

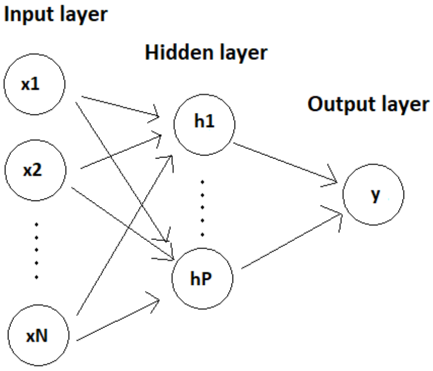


Figure 3: An illustration of an ANN with N inputs, one output, and one hidden layer with P nodes.

The output of an ANN with one hidden layer is given by

$$y = \varphi\left(\sum_{i=1}^{P} x_i w_i + w_0\right), \tag{2.19}$$

where φ is the activation function for the hidden layer, w_i are the weights for each node i and P is the number of nodes in the hidden layer. Simpler networks often use the hyperbolic tangent, $\varphi(a) = \tanh(a)$ as its activation function whereas deeper networks tend to use the rectifier function, $\varphi(a) = \max(0, a)$. It is however common to have the same activation function for all hidden layers. An illustration of the network described in Equation (2.19) is shown in Figure 3 where the input layer consists of N features obtained from the data set and P nodes in the hidden layer.

Training an ANN means optimizing the weights w to minimize the loss function E between the true values and our outputs, until convergence. Two of the most common optimization algorithms used are Stochastic Gradient Decent (SGD) and Adaptive Moment Estimator (ADAM), where ADAM is a further development of SGD [14, pp. 290–291]. The idea in both is to update the weights based on a random subset of the data, called a minibatch of size M. In SGD we update the weights w_k as

$$\Delta w_k = \frac{1}{M}\sum_{m=1}^{M} \Delta w_{mk} \qquad \text{and} \qquad \Delta w_{nk} = -\eta \frac{\partial E_n}{\partial w_k}, \tag{2.20}$$

where η is the learning rate and E_n the loss function for each feature. We will for simplicity reasons refer to $\frac{\partial E_n}{\partial w_k}$ as g_k in the following.

One potential drawback of SGD is that it can have a hard time minimizing the loss function if it gets stuck in a plateau. This problem is addressed by ADAM which instead makes use of two running averages of previous gradients when updating each weight,

$$\begin{aligned} m_k(t+1) &= \beta_1 m_k(t) + (1-\beta_1) g_k(t), \\ v_k(t+1) &= \beta_2 v_k(t) + (1-\beta_2)\left(g_k(t)\right)^2, \end{aligned}$$

Algorithm 6 Artificial Neural Network

```
1. Define all necessary hyperparameters.
     i. SGD: $\eta$ and $P$.
    ii. ADAM: $\eta, P, \beta_1, \beta_2$ and $\epsilon$.
2. Initialize variables to be updated.
     i. SGD: The weights $w_0$.
    ii. ADAM: The weights $w_0$, the running averages $m_0$ and $v_0$ and the initial
        gradient $g_k(t)$.
3. Repeat the following until $w$ converges.
   (a) Draw a minibatch of $M$ randomly chosen data points from the training
       set.
   (b) Compute the unique estimates for this minibatch
         i. SGD: Compute the gradient $g_k(t)$.
        ii. ADAM: Compute the gradient $g_k(t)$ and the moving averages $m_k(t)$,
            $v_k(t)$.
   (c) Update the weights $w$
         i. SGD: According to Equation (2.20).
        ii. ADAM: According to Equation (2.21).
```

where $m_k(t)$ is known as a momentum term. The weights are then updated according to

$$w_k(t+1) = w_k(t) - \eta \frac{\hat{m}_k(t+1)}{\sqrt{\hat{v}_k(t+1)} + \epsilon}, \tag{2.21}$$

where $\hat{m}_k(t+1) = \frac{m_k(t+1)}{1-\beta_1^t}$ and $\hat{v}_k(t+1) = \frac{v_k(t+1)}{1-\beta_2^t}$. This method thus introduces three new hyperparameters where β_1 and β_2 are exponential decay rates of the averages and ϵ is a small number such that we do not divide by zero.

2.3.4 Validation

Validating a model refers to measuring its performance on unseen data and comparing this to the performance on training data is how we detect overfitting. In Section 2.3.5 below, we discuss different metrics to evaluate models and their performance. These metrics are then used to find the ideal value of hyperparameters, and thus improve the model performance and reduce overfitting.

It is not possible to evaluate the model on the test set and then re-train it since this yields a biased model. Therefore, we construct a validation set by splitting the training set, which we then use to tune the hyperparameters. Once we are satisfied with the validation performance, we choose the best parameters and use those to train a final model based on both the training and validation sets before evaluating it on the test set. This whole process is called *model selection*. Including a validation set also has the additional benefit of lowering the risk of overfitting a model.

One common reason for overfitting is that we do not have enough data points to train the model on, especially after including a validation set. The K-fold cross-validation method helps to solve this by training and validating the model on different sections of the data and thus "generating" more data. The final training error is then the averaged training error of the different sections, see [16, pp. 242–249] for more details on this method. It is also advisable to use this method when tuning hyperparameters on a smaller data set.

2.3.5 Evaluation metrics

There are several kinds of evaluation metrics available when validating a model and depending on the purpose some might be more suitable than others. Having a metric that favors certain outcomes over others can for example be useful when designing a model to detect diseases since it often is better to over-diagnose than to under-diagnose.

In binary classification, the two outcomes are often referred to as "positive" and "negative", and four possible situations can arise upon classification. The model can correctly deduce the outcome and we then call it a *True Negative* (TN) or a *True Positive* (TP). If the outcome is wrongfully classified we either call it a *False Negative* (FN) or *False Positive* (FP). *Accuracy* is one of the most commonly used validation metrics and measures the proportion of predictions that were correctly classified. It is computed as

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}.$$

We can also compute the *Sensitivity* and *Specificity* from the the same quantities as,

$$\textbf{Sensitivity} = \frac{TP}{TP + FN}, \qquad \textbf{Specificity} = \frac{TN}{TN + FP}.$$

Sensitivity is the proportion of positive predictions correctly classified as Positive whereas sensitivity is the proportion of negatives correctly classified as negative. Outside of medical applications, these metrics are often referred to by other names, namely, Type I for specificity and Type II or recall for sensitivity [19, p. 149].

The *Receiver Operating Characteristic (ROC) curve* is used to visualize the trade-off between sensitivity and specificity. The area under the ROC curve is commonly referred to as the *Area Under the Curve (AUC)* and helps us to measure how close we are to an ideal ROC curve. An ideal model always correctly classifies its outcomes and we thus want both sensitivity and specificity to equal 1, yielding AUC = 1. The opposite case would be a purely random model which is equally likely to give a false positive as a true positive, corresponding to AUC = 0.5.

The F_1-*score* gives a more balanced overview of the model performance in cases of imbalanced classes and is defined as

$$\mathrm{F}_1 = 2 \cdot \frac{\underbrace{\frac{TP}{TP+FP}}_{\text{precision}} \cdot \underbrace{\frac{TP}{TP+FN}}_{\text{recall}}}{\underbrace{\frac{TP}{TP+FP}}_{\text{precision}} + \underbrace{\frac{TP}{TP+FN}}_{\text{recall}}}. \tag{2.22}$$

It is the harmonic mean between recall and precision, where precision is the proportion of true positives among all those classified as positives [34]. Including this metric can be beneficial since it can help determine if the model favors one of the two outcomes. A value close to 1 indicates a good model with even predictions whereas a value closer to 0 indicates an unsuccessful model.

We lastly mention a metric that is instead based on probability, namely the *log-loss*. This metric is sometimes referred to as the cross-entropy loss and is the same log-likelihood that we defined in Equation (2.18). We will however rewrite it in a more straightforward manner here and note that a lower $L_{\log}$ value indicates a better model,

$$L_{\log}(y,p) = -\log \Pr(y|p) = -\left(y\log(p) + (1-y)\log(1-p)\right),$$

where $y \in \{0,1\}$ and $p = \Pr(y=1)$.

3 Method Evaluation

3.1 Simulated data

Evaluating and comparing models is preferably done on simulated data since we then know what it is we want the models to detect. It is for example easier to evaluate a model's ability to detect a certain frequency if we know the frequency beforehand. If we wish to use our methods on real EEG data later, we need to simulate the signals such that they resemble true EEG signals. A framework to simulate such signals was presented in [4], where each signal $x(t)$ is constructed as a linear combination of the pure signal $x_s(t)$ and the noise $x_n(t)$,

$$x(t) = \sqrt{\frac{\lambda}{\lambda+1}} \frac{x_s(t)}{\|x_s(t)\|} + \frac{1}{\sqrt{\lambda+1}} \frac{x_n(t)}{\|x_n(t)\|}, \tag{3.1}$$

where λ is the associated Signal to Noise (SNR) factor. The noise $x_n(t)$ is constructed as a combination of white noise, alpha noise, and $\frac{1}{f}$-noise, whereas the pure signal $x_s(t)$ consists of Gaussian envelope transients. SNR describes the relation between the mass of $x_n(t)$ and $x_s(t)$. It is most often reported in decibels and a large SNR value corresponds to small amounts of noise.

We create two different data sets in order to answer the two research questions. The first data set is used to determine whether a stimulus, a signal, is present or not, and is referred to as the SignalDetection set, SigDet. We use Equation (3.1) to generate 8000 trials with $\lambda = 0$, which we view as the measurements before the stimulus is introduced, and 4000 × 2 trials with $\lambda > 0$ and phase-shifted $x_s(t)$s, which we view as our measurements with a stimulus. Both $x_s(t)$s are simulated with a frequency of 10 Hz and a sampling frequency of 256 Hz. Furthermore, the signals $x_1(t)$ and $x_2(t)$ are phase-coupled with a constant phase difference of $\frac{\pi}{4}$. The phase of $x_1(t)$ varies between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$, and the phase of $x_2(t)$ is consequently between $\frac{-\pi}{4}$ and $\frac{3\pi}{4}$. All

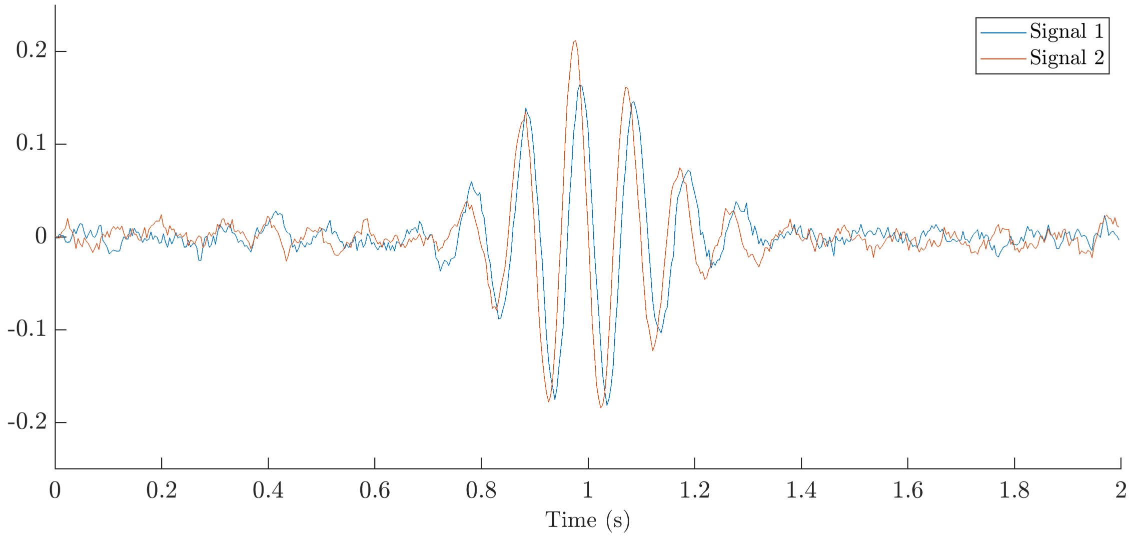


Figure 4: Two signals from the PhEst data set with SNR = 16 dB with different phase offsets.

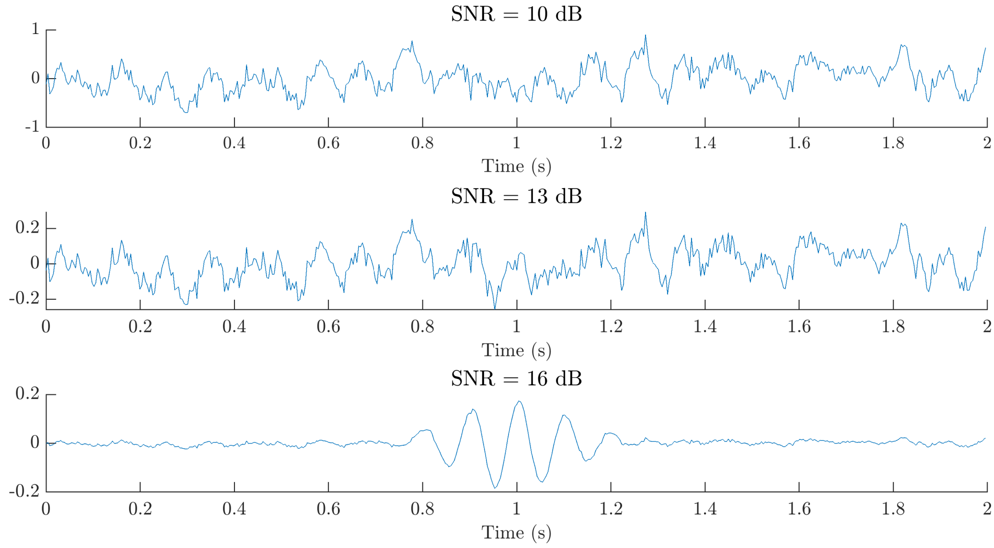


Figure 5: One signal from the SigDet data set for different SNRs.

four signals are two seconds long and the pure signal $x_s(t)$ is concentrated between 0.5 and 1.5 seconds in $x_1(t)$ and $x_2(t)$.

The second data set simulates the side scenario from research question two, and we refer to it as the PhaseEstimation set, PhEst. It is constructed by choosing 4000 phases θ_1 between $\frac{-\pi}{2}$ and $\frac{\pi}{2}$ and 2000×2 associated phases $\theta_2 = \theta_1 \pm \frac{\pi}{4}$. These phases are then used to generate two sets of $x_s(t)$s with the same underlying frequency and sample frequency as in SigDet, which we combine with noise to get 2000×2 realizations of (3.1). The phase offsets are chosen so that the first 2000 realizations have a positive phase difference $\Delta\theta > 0$, and the last 2000 have a negative phase difference $\Delta\theta < 0$, which corresponds to the two sides. An illustration of two signals with a positive phase difference can be found in Figure 4.

Finally, we generate three variations of each data set by changing the SNR factor in Equation (3.1), in order to determine how robust the methods are to noise. The different SNR factors are 10 dB, 13 dB and 16 dB, and a signal from the SigDet data set is shown in Figure 5 for the different SNRs. We expect the methods to work perfectly for 16 dB since the pure signal $x_s(t)$ is clearly visible. It is still possible to see traces of $x_s(t)$ for 13 dB and the methods should therefore manage to identify its characteristics, whereas the methods should struggle in the 10 dB case.

3.2 Stationary methods

3.2.1 Method

The methods in this section assume stationarity and we therefore segment the data into shorter, overlapping segments. This segmentation gives us shorter signals that exhibit more stationarity by not varying as much over time. We divide both signals into 0.5-second segments with a 50% overlap. We then compute the cross-spectrum and the respective auto spectrum of each segment, using $K = 4$ Hanning windows, and all segment spectrums are thereafter individually summed. The final coherence

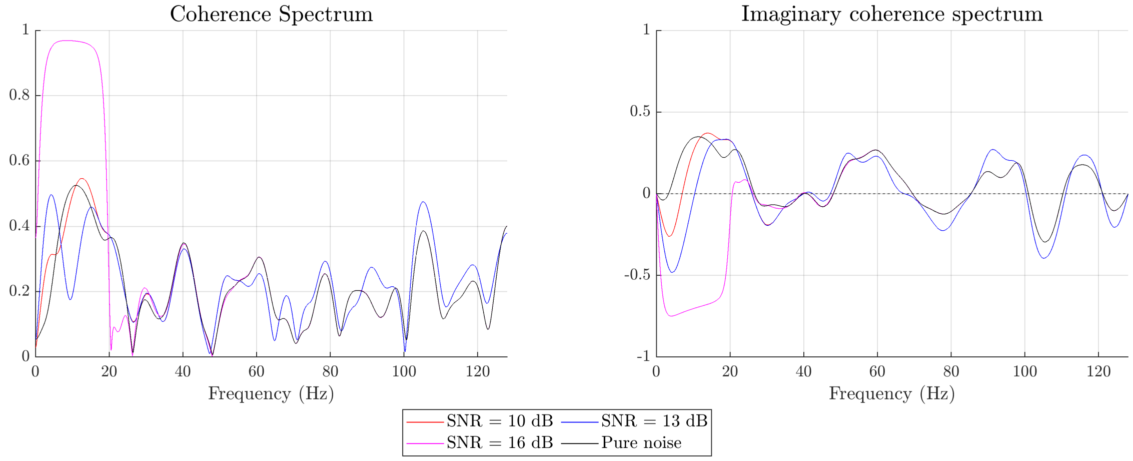


Figure 6: The coherence spectrum and imaginary coherence spectrum of one trial from the SigDet data set, simulated with different SNRs.

and imaginary coherence are then computed as in Equation (2.6) and (2.8) based on the summed cross- and auto spectrum.

3.2.2 Evaluation

Applying both methods to the SigDet data set shows that the models are able to distinguish the presence of the pure signal $x_s(t)$ when SNR $=$ 16 dB, see Figure 6. While the methods are able to detect a correlation in the 0 to 20 Hz frequency band for lower SNR, the performance is significantly degraded. Moreover, the coherence spectrum on the left appears to be more sensitive to noise than the imaginary coherence spectrum on the right. However, the strength of the correlation is easier to interpret in the coherence plot and it can thus, as expected, be wise to include both.

Both methods are similarly able to identify a correlation when evaluated on the PhEst data set for higher SNR. The sign of the phase difference does not significantly affect the coherence spectrums as can be seen in Figure 7. Note that the energy in the 0

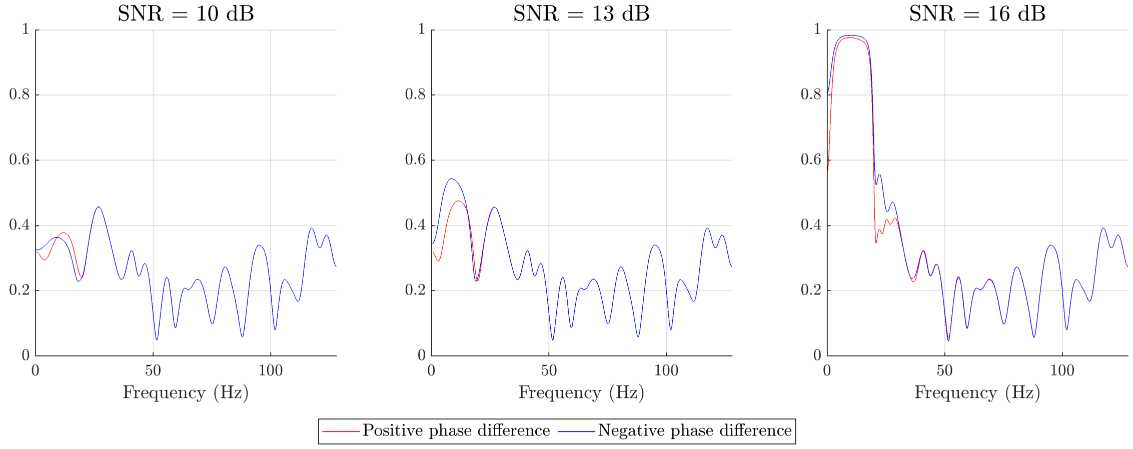


Figure 7: The coherence spectrums of two trials with opposite phase differences and the same underlying noise at different SNRs, from PhEst.

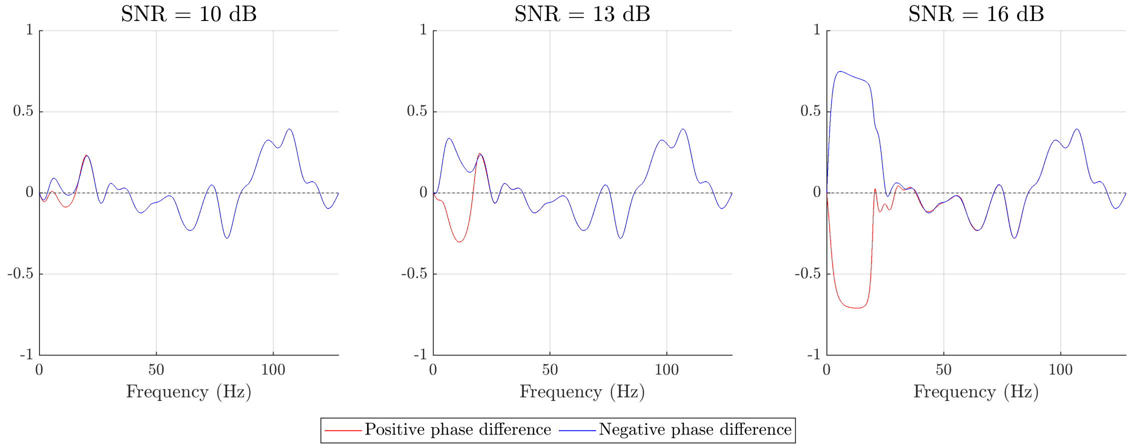


Figure 8: The imaginary coherence spectrums of two trials with opposite phase differences and the same underlying noise at different SNRs, from PhEst.

to 20 Hz region increases with SNR, as is to be expected. On the other hand, the imaginary coherence spectrums of the same trials are able to distinguish between the sign of the phase difference, see Figure 8. The two trials appear to mirror each other in the frequency band of interest and this difference is still evident for lower SNRs. This suggests that the sign of imaginary coherence can be used for answering the second research question. Nonetheless, the coherence spectrum is more suitable for determining the strength of correlation since it captures the full energy of the signal.

Both stationary methods can be used in EEG analysis and it is advisable to segment the data in order to obtain a more stationary behavior. The coherence spectrum is better at distinguishing the strength of correlation, whereas the imaginary coherence spectrum can distinguish between the sign of phase differences. We are thus able to use both methods to answer research question one and the imaginary coherence spectrum for research question two. There might be more benefits with the imaginary coherence spectrum but we need a different data set to evaluate since we have not included volume conditions in the simulations.

3.3 Non-stationary methods

3.3.1 Method

There is no need to segment the data before evaluating the methods in this section since we do not assume stationarity. We compute the PLV, iPLV, and ciPLV by designing a band-pass filter between 8 and 12 Hz and filtering each signal before extracting the respective phase difference between $x_1(t)$ and $x_2(t)$ after applying the Hilbert transform. Now we compute all metrics across both trials and time in order to assess the methods' performances. The summations in Equations (2.9), (2.10), and (2.11) are thus performed twice. Both filters for PLV_{amp} are inspired by [18] and we set the lower-frequency filter to be between 9 and 11 Hz, and the higher-frequency filter to be between 50 and 70 Hz.

The SSD algorithm yields an unknown number of components K since the algorithm is repeated for as long as NMSE $> th$. Our PLV computations are however only based on the component with the most power within the alpha region since we know that the pure signal $x_s(t)$ has a frequency of 10 Hz. In addition, we set a maximum number of components that the SSD can produce, $K = 10$, in order to accelerate the computations. After identifying the component, we compute the PLV, iPLV, and ciPLV as described above, excluding the filtering process. Finally, we compute the cross-spectrum of the entire signal and extract its imaginary component in order to get PLI and wPLI. The PLIs are only computed individually for each trial.

3.3.2 Evaluation

Normal PLV appears to identify phase-coupling during the time period of interest when computed on the SigDet data set, as can be seen in Figure 9. Without $x_s(t)$, the PLV is effectively constantly zero while with $x_s(t)$, there is a clear increase for all three cases of SNR between 0.5 and 1.5 seconds. As we decrease SNR, we see a decrease in PLV strength which is reasonable, but the decrease between 13 and 10 dB is rather rapid, suggesting that the method may struggle when more noise is present. Regardless, all PLV differences are deemed significant with a p-value $\ll 0.01$.

Using SSD instead of filtering yields a stronger and smoother PLV as can be seen by comparing Figure 9 and 10. We see that for SNR = 16 dB, $\text{PLV}_{\text{SSD}} = 1$ during the time that $x_s(t)$ is present, whereas normal PLV has an oscillatory behavior with a maximum value of approximately 0.7. Similarly to normal PLV, we see a weakened strength in PLV_{SSD} as SNR decreases, but the maximum value of PLV_{SSD} is still higher for the same SNR, compared to normal PLV.

There are some limitations with the simulated data set, which affect our evaluations of PLV_{amp}, iPLV, and ciPLV. There are no high-frequency components in this data set, which causes no significant results for PLV_{amp}. There is no difference between $x_s(t)$ being included or not and all PLV_{amp}s very small and there is no difference. Furthermore, our iPLVs and ciPLVs show great similarity when obtained through both

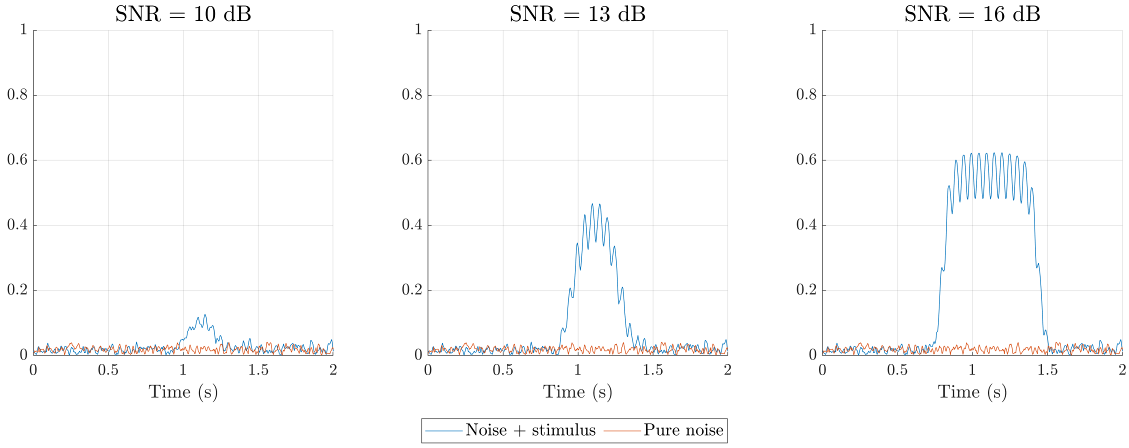


Figure 9: PLV computed across all SigDet trials, for different SNRs as well as pure noise for comparison.

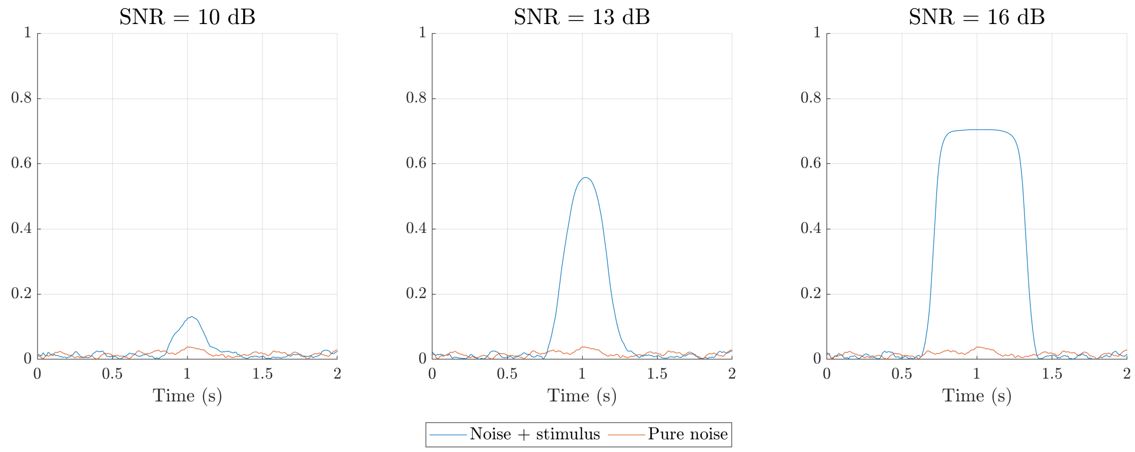


Figure 10: $\mathrm{PLV}_{\mathrm{SSD}}$ computed across all SigDet trials, for different SNRs as well as pure noise for comparison

filtering and SSD. This is most likely due to the fact that we have not introduced any volume conduction in our simulated data set, which is what should distinguish the two according to Section 2.2.2. However, the sign of $\mathrm{iPLV}_{\mathrm{SSD}}$ and $\mathrm{ciPLV}_{\mathrm{SSD}}$ is affected by the sign of the phase difference in the PhEst data set, as can be seen in Figure 11. The difference is in addition clearly visible for the lowest SNR, but $\mathrm{iPLV}_{\mathrm{SSD}}$ and $\mathrm{ciPLV}_{\mathrm{SSD}}$ are then nearly identical. Using the imaginary-based PLV can therefore help us answer the second research question. This sign difference is also present with normal iPLV or ciPLV but it is more subtle.

All methods are more sensitive to the sign of the phase difference when we compute them over time instead of trials. We can divide the PhEst data set into three cases; trials with positive phase differences, negative phase differences, or a combination of the two, were the first case corresponds to the signals in the SigDet data set. When the sign of the phase difference $\Delta\theta$ is kept constant for SNR = 13 dB, we see that all methods, except PLI, manages to distinguish between $x_s(t)$ being present or

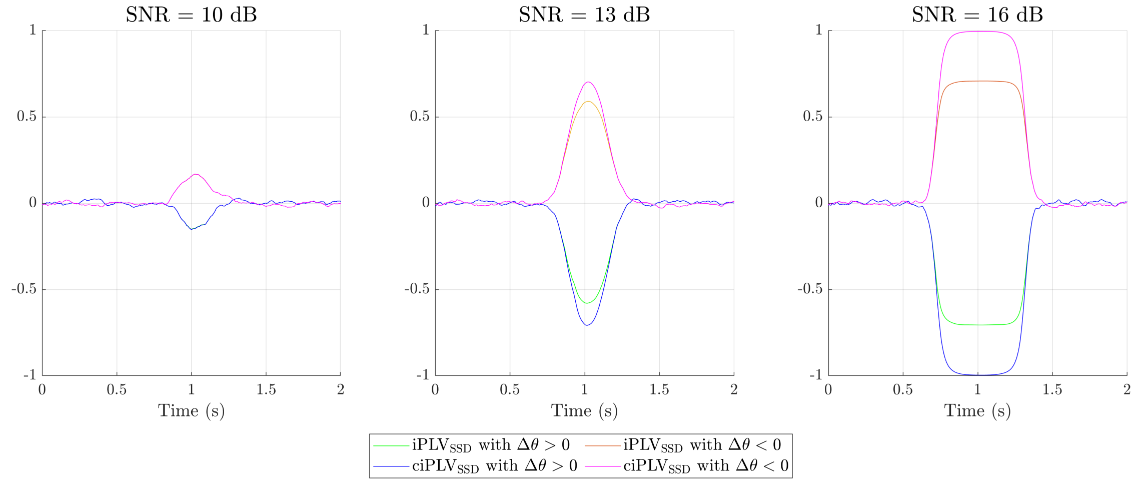


Figure 11: $\mathrm{iPLV}_{\mathrm{SSD}}$ and $\mathrm{ciPLV}_{\mathrm{SSD}}$ computed across all PhEst trials with the same signed phase difference for different SNRs.

Table 2: The mean of all nine methods, and their standard deviations in parenthesis, when computed over time, based on the PhEst data set with SNR = 13 dB. A statistically significant deviation from noise is denoted by an asterisk (*).

Metric	**Trials with $+\Delta\theta$**	**Trials with $-\Delta\theta$**	**All trials**
sPLI	-0.107 (0.994)*	0.141 (0.990)*	0.017 (1.00)*
PLI	0.098 (0.073)	0.100 (0.075)	0.099 (0.074)
wPLI	0.482 (0.252)*	0.494 (0.254)*	0.488 (0.253)*
PLV	0.263 (0.131)*	0.263 (0.134)*	0.263 (0.132)*
iPLV	-0.087 (0.184)*	0.090 (0.188)*	0.001 (0.206)
ciPLV	-0.089 (0.188)*	0.092 (0.193)*	0.001 (0.211)
PLV_{SSD}	0.263 (0.131)*	0.261 (0.132)*	0.262 (0.131)*
iPLV_{SSD}	-0.084 (0.188)*	0.086 (0.188)*	0.002 (0.206)
$\text{ciPLV}_{\text{SSD}}$	-0.086 (0.192)*	0.088 (0.193)*	0.001 (0.211)

not, as can be seen in Table 2. The sign of the phase difference only influences the sign of the imaginary-based means, which in turn indicates that they can be used to answer the second research question. On the other hand, when the sign of $\Delta\theta$ varies, all imaginary-based methods struggle to detect the stimulus. However, we have an even number of positive and negative phase differences and it is thus probable that the values are canceled out in the means.

The only difference between SNR being 13 or 16 dB is the significance of PLI, and the means for the 16 dB case are thus not shown. In contrast, only the imaginary-based methods yield significant results when we set SNR to 10 dB and keep the sign of $\Delta\theta$ constant, as can be seen in Table 3. Moreover, only the sPLI manages to distinguish between noise and stimulus when $\Delta\theta$ varies. Finally, the PLI methods are generally more sensitive to noise than the PLV methods, but the sPLI can distinguish between stimulus and noise for all SNR levels as well as the sign of the phase difference when SNR $\geq$ 13 dB.

Table 3: The mean of all nine methods, and their standard deviations in parenthesis, when computed over time, based on the PhEst data set with SNR = 10 dB. A statistically significant deviation from noise is denoted by an asterisk (*).

Metric	**Trials with $+\Delta\theta$**	**Trials with $-\Delta\theta$**	**All trials**
sPLI	-0.024 (1.00)	0.045 (0.999)	0.011 (1.00)*
PLI	0.097 (0.072)	0.098 (0.074)	0.098 (0.073)
wPLI	0.399 (0.236)	0.408 (0.236)	0.403 (0.236)
PLV	0.247 (0.125)	0.248 (0.125)	0.248 (0.125)
iPLV	-0.015 (0.191)*	0.020 (0.196)*	0.002 (0.194)
ciPLV	-0.015 (0.195)*	0.020 (0.200)*	0.002 (0.198)
PLV_{SSD}	0.246 (0.124)	0.246 (0.128)	0.246 (0.126)
iPLV_{SSD}	-0.013 (0.192)*	0.017 (0.198)*	0.002 (0.195)
$\text{ciPLV}_{\text{SSD}}$	-0.014 (0.196)*	0.017 (0.201)*	0.002 (0.199)

The methods evaluated in this section can be used for EEG analysis and all variations of the PLV can be computed across both time and trials. The methods do however appear to be more sensitive to noise and the sign of the phase difference when averaging PLV over time. Identifying whether a stimulus is present is easier when the sign of the phase difference is consistent, but the methods are able to detect a correlation in other cases too. Moreover, the imaginary PLVs are able to detect the sign difference and thus answer the second research question in the affirmative. In addition, the PLV methods are more robust to noise than the PLI methods. Finally, using SSD instead of filtering is not only easy to implement, but it also yields better results and is therefore preferred.

3.4 Machine Learning

3.4.1 Method

We train all models on both data sets, SigDet and PhEst data sets in order to address the two research questions. All classifications algorithms discussed in Section 2.3 take scalar input features and we extract 14 such features. The PLV and PLI computations are all computed across time but averaging the coherence- and imaginary coherence spectrum in a similar manner is not ideal. As we saw in Section 3.2, these methods only generate relevant information in the frequency band of interest and we therefore sum the power between 8 and 12 Hz, which is inspired by [12]. We also compute the power of the PSD from the chosen SSD component, in the same interval, and record its maximum value. Both of these features are done separately for $x_1(t)$ and $x_2(t)$ and the absolute value of their difference is then used as the final feature.

We also include features that consider the sign of the imaginary coherence and PLI in the PhEst models. The sign of the imaginary coherence feature is obtained by integrating the signed power of $i\mathcal{C}_{xy}$ between 8 and 12 Hz. Finally, PLV_{amp} is not included in either model since we do not have any high-frequency components in our data.

In order to improve the performance of our models, we evaluate the importance of different features. Correlated features can negatively impact the model performance and we thus check for features with strong correlations. This is especially important in our case since we include different variations of the same method, such as PLV and PLV_{SSD}. The correlated features are then removed separately and we use the evaluation metrics from Section 2.3.5 to determine which feature removal yields the best validation performance. Next, we check the importance of different features to see if our predictions improve.

Once we are satisfied with the number of features we begin to tune the hyperparameters for the RF and ANN models by making use of a 5-fold cross-validation. We tune three hyperparameters for the RF model: the number of trees B, the number of features considered for each split m, and the maximum number of splits S, whereas we tune four hyperparameters for the ANN model: The number of hidden layers H, the choice of activation function φ, the choice of optimization algorithm Opt and the

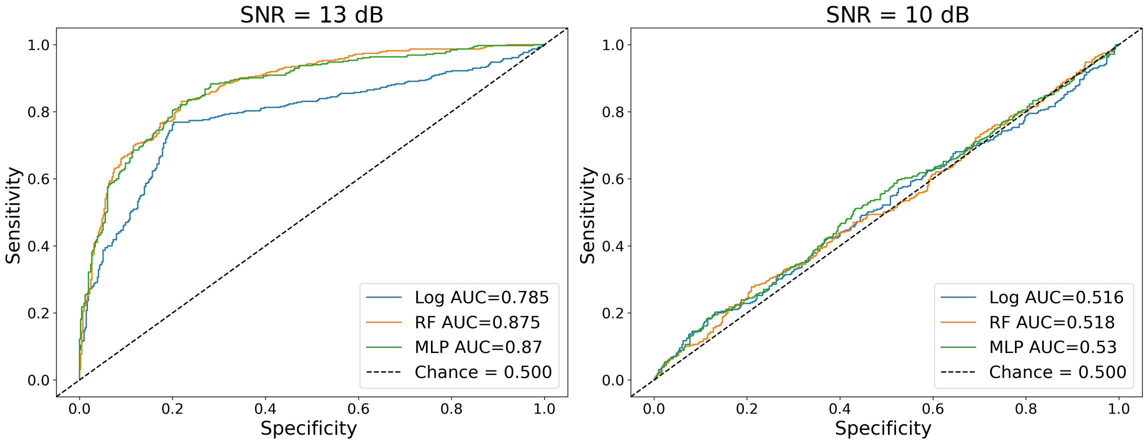


Figure 12: ROC curves for the PhEst based models for different SNR.

regularization strength r. The best combination of hyperparameters is then used to train the final models which are evaluated on the test set.

3.4.2 Evaluation

The final hyperparameters of all models are shown in Table 4 for SNR $=$ 10 and 13 dB. There are strong similarities between both data sets and the two SNR levels 10 and 13 dB. The main difference is the depth of the RF models, where both PhEst based models contain more splits than the models based on SigDet set. However, the second research question is expected to be more complex and it is therefore reasonable that it requires more elaborate models. The values of the hyperparameters for the SNR $=$ 16 dB models are similar and are therefore omitted.

All SigDet based models perform well on both the validation and test sets. The validation metrics on the test set for SNR $=$ 10 and 13 dB are shown in Table 5 where we for example see that the accuracy is greater than 95.9% for all models and both SNR. The RF model appears to be the best model for both SNR cases, and this model has a 100% accuracy when SNR $=$ 13 dB. The corresponding ROC curves are shown in Figure 18 in the appendix and are nearly perfect. The feature importance

Table 4: The final choice of hyperparameters for the sets of models when SNR equals both 10 and 13 dB and $\boldsymbol{N}$ is the total number of features.

	Data set				**Hyperparameter**				$\boldsymbol{N}$
		B	m	S	H	φ	Opt	r	
SNR $=$ 10 dB	PhEst	5	2	64	5	relu	ADAM	10^{-5}	8
	SigDet	5	2	8	5	tanh	ADAM	10^{-3}	7
SNR $=$ 13 dB	PhEst	5	4	128	5	relu	ADAM	10^{-5}	8
	SigDet	5	2	16	5	tanh	ADAM	10^{-5}	7
		RF paramaters			ANN paramaters				

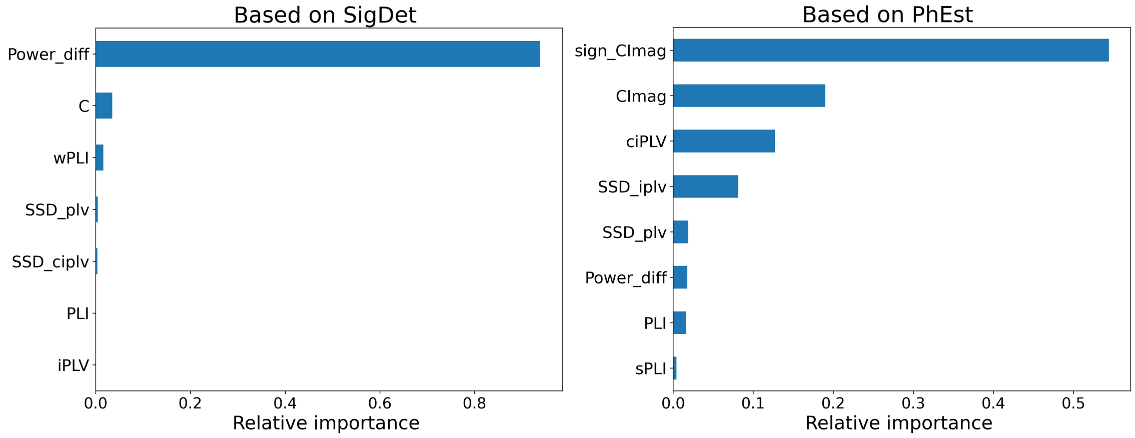


Figure 13: Feature importance for both RF models with SNR = 13 dB.

of the RF model for SNR = 13 dB is shown on the left in Figure 13 which tells us that the $\text{SSD}_{\text{Power}}$ is the most important feature of the model. The same applies in the SNR = 10 dB case and removing this feature in both cases worsens the results drastically. This is reasonable considering that the average $\text{SSD}_{\text{Power}}$ for pure noise is 100 times greater than that for SNR = 10 dB. This large difference might be due to our simulated data being constructed in a way that makes it especially well suited for the SSD algorithm, but it is regardless not likely to be as pronounced on real data.

The PhEst based models do not perform as well as those based on SigDet, as can be seen in Table 5. The RF model performs best for SNR = 13 dB with an accuracy of 79.5%, and the ANN model performs best for SNR = 10 dB with a 53.8% accuracy. Furthermore, the ROC curves in Figure 12 show that the RF and ANN models perform relatively similarly for both SNRs whereas the logistic model has the worst trade-off between sensitivity and specificity. The imaginary coherence spectrum and the imaginary PLV methods are as expected the strongest indicators of which signal is leading, see the right figure in Figure 13. Finally, when SNR = 16 dB all methods have perfect predictive capabilities which is why we do not discuss it further.

Table 5: The validation metrics of all models on the test set, for SNR = 10 dB and SNR = 13 dB. The best results are written in bold.

SNR	**Metric**	**Based on SigDet**			**Based on PhEst**		
		Log	**RF**	**ANN**	**Log**	**RF**	**ANN**
13 dB	L_{log}	1.30	**0.00**	0.410	7.47	**7.08**	7.30
	AUC	0.991	**1.00**	0.998	0.785	0.850	**0.871**
	F_1	0.962	**1.00**	0.988	0.774	**0.789**	0.773
	Accuracy	0.963	**1.00**	0.989	0.784	**0.795**	0.789
10 dB	L_{log}	1.40	**0.410**	0.648	16.7	17.0	**16.0**
	AUC	**0.996**	0.994	0.996	0.517	0.521	**0.530**
	F_1	0.960	**0.988**	0.981	**0.533**	0.498	0.516
	Accuracy	0.959	**0.988**	0.981	0.520	0.509	**0.538**

The three classification algorithms presented here are all suitable classifiers for EEG data. It does however not appear to be any distinct differences between the models which suggests that the obtained features are suitable for EEG analysis. The ANN model might be better at handling more complex tasks, but the RF model performs very well as well whereas the logistic model struggles the most. Furthermore, one feature is an exceptionally strong indicator of whether a signal $x_s(t)$ is present or not, and we therefore have a very high accuracy rate for SNR = 10 dB. This feature may however only yield such strong results on simulated data due to the way it reconstructs the signals. The imaginary-based methods are the greatest at detecting the sign of the phase difference, but the models do not have a high accuracy when classifying the difference for lower SNR.

4 Real EEG Examples

4.1 Visual stimulus

Each EEG recording is 15 seconds long, during five of which, the subject was exposed to flickering lights of different frequencies. The subject was sitting in a silent room with their eyes closed throughout the recording. One recording was made for each frequency, for a total of 10 recordings, with a sampling frequency of 256 Hz. The lights were generated through a Grass Photic stimulator Model PS22C and the data was recorded using a Neuroscan system with a digital amplifier (SYNAMP 5080, Neuro Scan, Inc.) and 21 electrodes [31].

4.1.1 Methodology

Only the first research question is relevant to this data set, and we therefore divide the data into three subsets; before, during, and after the flickering lights are present. Determining if the visual stimulus is present is done by comparing the five seconds before the lights are flickering with the five seconds that they are. Visual information is processed in the occipital lobe at the back of the brain [28, p. 260], and we therefore choose all combinations of electrode pairs such that electrode 02 or 01 are always present, see Figure 19 in the appendix for electrode placement. We consider the measurements where the lights flicker at 12 and 15 Hz since these frequencies are easier to detect.

The signal processing methods are applied in a similar manner to that described in Section 3.2.1 and 3.3.1, and no classification algorithms are considered. We segment the data into 1.5-second segments with a 50% overlap before computing the coherence- and imaginary coherence spectra. The two recordings of lights flickering at 12 and 15 Hz are treated separately since the spectra are frequency-based, whereas the phase-based methods can be computed with respect to electrode pairs from both. Nonetheless, the filters for the PLV computations differ between the two since they need to be centered around the frequency at which the lights flicker. Similarly, we choose the SSD component with the largest amount of power in 4 Hz wide frequency bands centered at 12 and 15 Hz, respectively.

4.1.2 Results

Computing the spectra for both flickering lights of 12 and 15 Hz yields similar results and we thus only discuss the 15 Hz further. The imaginary coherence spectrum detects volume conduction and it is therefore possible to distinguish true connectivity from false. This can for instance be seen when computing both coherence spectra between the electrode pair O1-O2. These electrodes are placed next to each other and we thus expected them to naturally have a high correlation, which is confirmed by the coherence spectrum (left) in Figure 14. During the time that the lights flicker, we see a peak around 15 Hz but it is hardly distinguishable from other frequencies either before or during the flickering. Furthermore, there are no peaks, nor strong correlations in the imaginary coherence spectrum (right) in Figure 14 implicating that the recordings are affected by volume conduction.

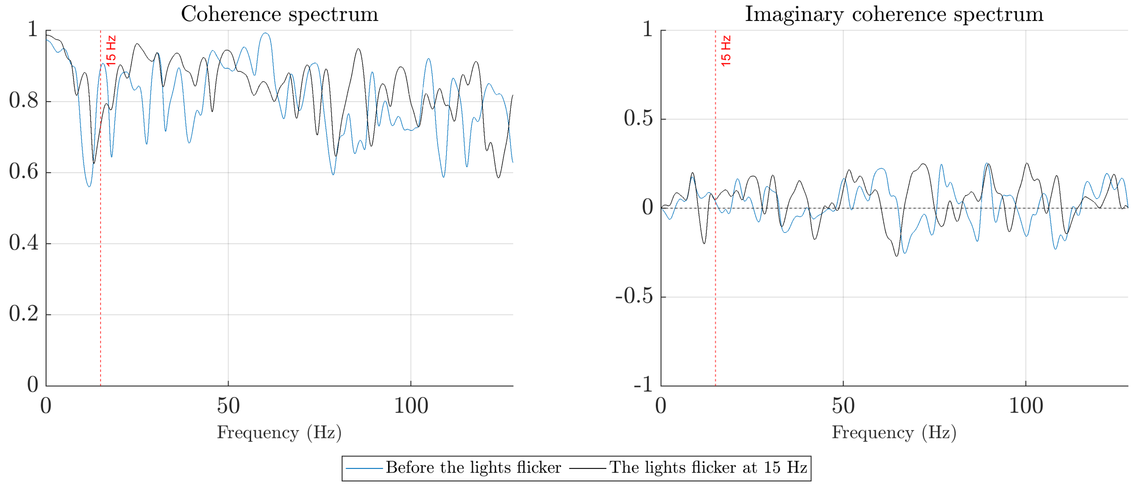


Figure 14: The coherence spectrum and imaginary coherence spectrum between the electrode pair O2-O1, before and during the lights flicker at 15 Hz.

An additional benefit of identifying volume conduction is that the peak in the imaginary coherence spectrum is easier to find. This can, for example, be seen in Figure 15 where it is easier to detect the frequency of correlation between the electrode pair O2-F8 in the imaginary coherence spectrum (right) than the coherence spectrum (left). The imaginary coherence spectrum also makes a clear distinction between the time before and during the stimulus is present.

The non-stationary methods do not treat each electrode pair separately, and we can therefore not know which pair exhibits the strongest connectivity. The methods can however successfully distinguish between the time before and after the lights are introduced. The imaginary-based PLVs are able to differentiate between the two, whereas the normal PLVs are not. These results are not surprising since we know from Figures 14 and 15 that volume conduction is present. Furthermore, the iPLV and ciPLV are nearly identical and there is no major difference between them being obtained

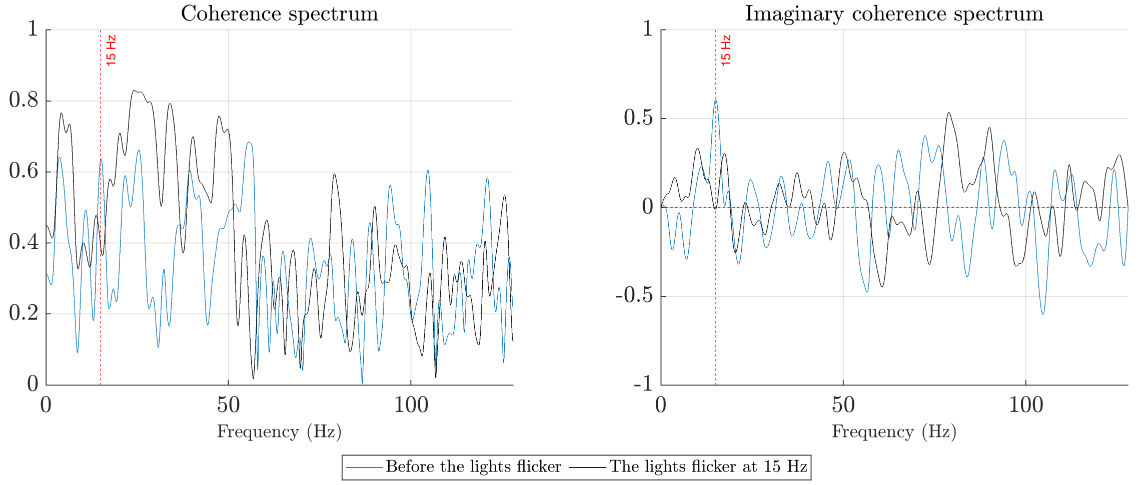


Figure 15: The coherence spectrum and imaginary coherence spectrum between the electrode pair O2-F8, before and during the lights flicker at 15 Hz.

through filtering or the SSD algorithm.

The imaginary coherence spectrum and the PLV methods are all able to distinguish between the time before and after a visual stimulus of flickering lights is introduced. The imaginary coherence spectrum can detect volume conduction and is thus especially well suited for identifying the frequency at which the lights flicker. All imaginary-based methods are in addition able to determine if the stimulus is present, whereas the non-imaginary methods are not as successful. Lastly, the electrode pairs that exhibit the greatest connectivity all have one electrode in the occipital lobe and one in the right frontal lobe, suggesting that these brain regions interact when processing the flickering lights.

4.2 Auditory stimulus

This data set consists of approximately nine-second-long EEG measurements of six different subjects, numbered 10-15 taken from a larger dataset, where each subject is presented with a word in either their left or right ear. Each subject was not presented with any stimuli during the first two seconds of recording and the word was presented after approximately 2.6 seconds through in-ear headphones (SONY MDR EX650AP). A more thorough explanation of the data set can be found in [27]. Originally the sampling frequency was set to 1000 Hz but it was later down-sampled to 256 Hz, and the data was then preprocessed with a high-pass filter of 0.1 Hz, and line noise at 50 Hz was reduced. Trials containing artifacts were removed through visual inspection and independent component analysis, which identifies ocular and muscle artifacts. The final number of trials per subject varied between 343 and 370 trials. Finally, the data was recorded using a NeruoScan SynAmps RT (Compumedics) amplifier, Curry 7 software, and 64 electrodes placed according to the 10% system, covering the 10/20 area, in an EasyCap.

4.2.1 Method

Similarly to Section 3.4.1, we train two sets of models, each corresponding to one of the two research questions. Furthermore, each set is trained individually for all subjects since EEG recordings can vary greatly between people. The first set of models is referred to as the *significance models* and is designed to distinguish between silence and a word being said. The second set is referred to as the *side models*, and is meant to determine which ear the word was said into. We know that the word was said sometime between 2.6 and 5.6 seconds, and we therefore choose the entirety of the three seconds as our stimulus subset to guarantee that word is said. The silence subset consists of the first two seconds of the recording since no stimuli were present.

Sound is processed in the auditory cortex [28, p. 309] and we therefore choose recordings from 15 electrode pairs located at the center of the scalp and the temporal lobe to train the significance models on. On the other hand, the side-based scenarios are trained on data from ten electrode pairs, all located close to the ears. Each pair is chosen to be symmetrically placed on the scalp, for example, T7-T8 and P1-F2. The pairs and their locations are shown in Figure 20 in the appendix. The extraction of features is nearly identical to that described for the simulated data sets throughout Sections 3.2 to 3.4, including the segmentation length of 0.5 seconds and the filter bandwidths since we limit ourselves to only investigating brain activity in the alpha region. We do include the PLV_{amp} to investigate potential connectivities between alpha and gamma waves. Both sets of models are evaluated on the validation set for all subjects and the best subject's models are then evaluated on the test set.

4.2.2 Results

Training any model on all subjects does not increase its performance compared to only training it on the best subject. This is however not surprising since we know that EEG measurements vary between individuals. Furthermore, there is a seven percentage point difference between the best and worst subject which further strengthens this reasoning. Only the best subject for both sets of models will thus be presented and the value of their hyperparameters can be found in Table 7 in the appendix.

The subject yielding the best set of significance models is subject 13, where the best one is the RF model. This model has an accuracy of 68.7% on the test set, as can be seen on the left in Table 6. Additionally, the ROC curves on the left in Figure 16 show that the trade-off between sensitivity and specificity is approximately 0.745 for both the RF and ANN models which, in addition to a good F_1-score, suggest that the model performs well.

The side models perform the best when trained on subject 11, with the RF model having the greatest model performance. The validation metrics on the test set can be seen on the right in Table 6, where an equivalent accuracy and F_1-score indicate a balanced model. An accuracy of 55.1% is better than chance, and that combined with an AUC of 0.573, suggests that at least some of the features are relevant for distinguishing the side. The ROC curves for the three side models are shown on the right in Figure 16. The figure reaffirms the RF model as slightly superior to the other two since the RF curve has the best trade-off between sensitivity and specificity.

The feature importance plot for the significance RF model is shown on the left in Figure 17, from which we can conclude two important things. The SSD algorithm

Table 6: Validation metrics computed on the test set for the two best subjects, for each set of models on real EEG data with auditory stimulus. The best results are written in bold.

Metric	**Significance models**			**Side models**		
	Log	**RF**	**ANN**	**Log**	**RF**	**ANN**
L_{log}	11.5	**10.8**	11.1	16.0	**15.5**	15.7
AUC	0.735	0.743	**0.746**	0.555	**0.573**	0.555
F_1	0.686	**0.700**	0.692	0.537	**0.551**	0.514
Accuracy	0.666	**0.687**	0.680	0.538	**0.551**	0.545
	Based on subject 13			Based on subject 11		

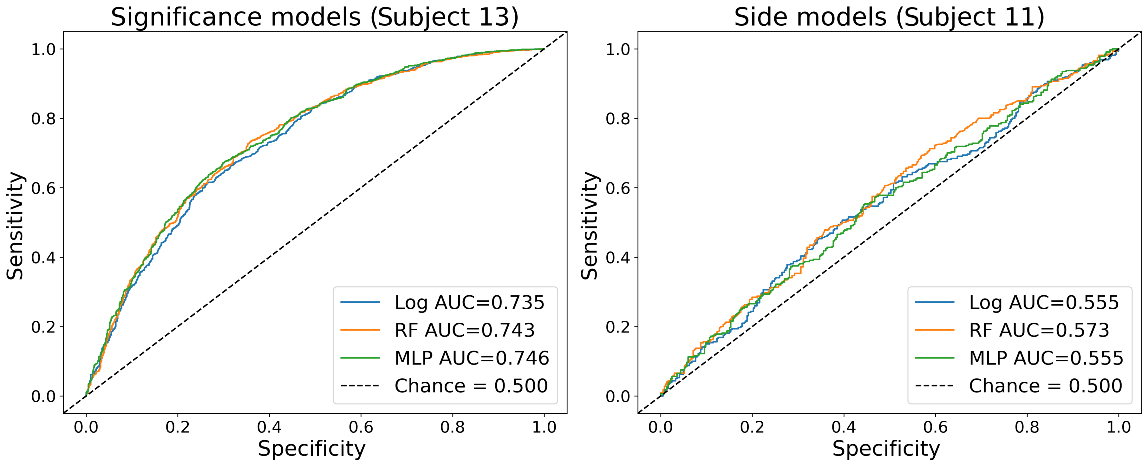


Figure 16: The ROC curves for all models on the auditory EEG data.

yields the most important feature $\text{SSD}_{\text{Power}}$, though not as important as for the simulated SigDet data set. Secondly, the importance of PLV_{amp} suggests that there is an interaction between alpha and gamma brain waves when subject 11 processes sound. It could be interesting to combine these features in a feature since they have the potential to partner well.

The side RF model does not have a singular feature that is more important than the others, as can be seen in its feature importance plot on the right in Figure 17. The conclusions from this plot are not as reliable as in the previous paragraph since the model only has an accuracy of 55.1%. Regardless, most features have approximately the same relative importance on the side model. The plot does however suggest that the sign of the PLI and imaginary coherence are not as important as the metrics themselves, contradicting our expectations from Section 3.4.1. On the other hand, the best methods for detecting the side are the imaginary methods which are in accordance with previous parts of this thesis.

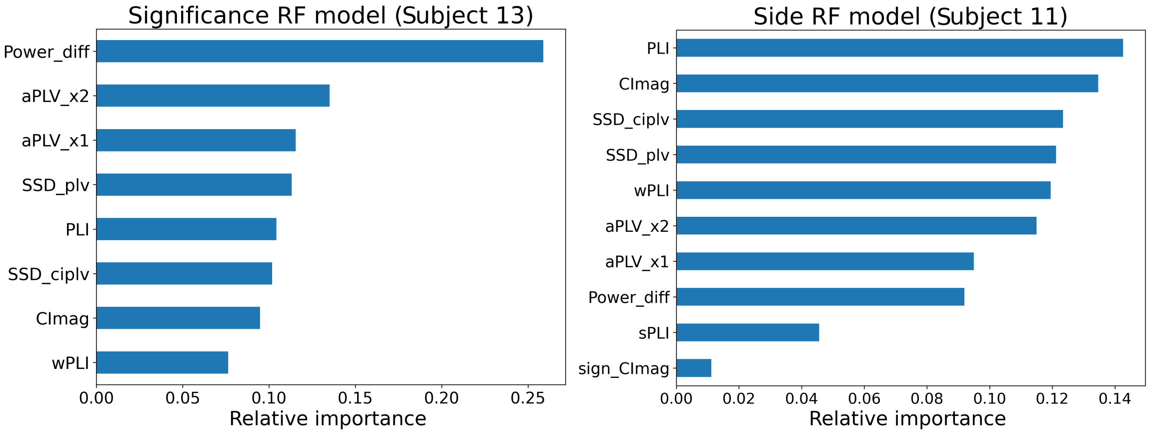


Figure 17: Feature importance for the RF models on auditory EEG data.

Lastly, the model performances are similar within the two sets, indicating that we have included relevant features in our classification algorithms. To clarify, it would have been difficult to deduce whether the final results were due to the selected features or model selection if the models' prediction capabilities differed greatly. We can thus conclude that the signal processing methods used to extract the features are suitable for real EEG analysis since the evaluation metrics are roughly the same for all sets of models.

5 Conclusion

EEG analysis is an important tool in both research and medicine, and this thesis has evaluated and discussed several suitable methods in relation to real and simulated data. The main focus has been on methods that address volume conduction since it can have negative effects on EEG analysis, as well as non-stationary methods due to the noisiness of EEG recordings. Most non-stationary methods presented are variations of the PLV but the stationary coherence spectrum has also been evaluated.

The findings of this thesis suggest that some methods might be more suitable than others depending on the purpose of the analysis. The imaginary coherence spectrum is, for example, efficient at removing volume conduction effects, and is therefore better suited for EEG analysis than the normal coherence spectrum. There is no clear difference between the iPLV and ciPLV, and both are suitable for detecting the sign of the phase difference. They do however struggle if the sign alternates between trials. The PLV_{amp} is able to detect coupling between alpha and gamma waves when a subject processes sound, and it is therefore highly suitable to include it in most EEG analyses. Furthermore, using the SSD algorithm to decompose a signal instead of filtering has shown to always yield smoother, clearer, and more interpretable PLVs. In addition, the most important feature for classifying whether a stimulus is present is based on the SSD algorithm, and it would consequently be interesting to investigate this algorithm further, by for example combining it with PLV_{amp}.

Conclusively, it is easier to detect the presence of an auditory stimulus compared to determining the part of the brain that processed it first. The methods in this thesis yield a maximum accuracy of 68.7% for the first scenario and 55.1% for the second.

References

[1] Abhang, P. A., Gawali, B. W., and Mehrotra, S. C., “Chapter 2 - Technological Basics of EEG Recording and Operation of Apparatus,” in *Introduction to EEG- and Speech-Based Emotion Recognition*, Academic Press, 2016, pp. 19–50, ISBN: 978-0-12-804490-2. DOI: 10.1016/B978-0-12-804490-2.00002-6.

[2] Aydore, S., Pantazis, D., and Leahy, R. M., “A note on the phase locking value and its properties,” *NeuroImage*, vol. 74, pp. 231–244, 2013. DOI: 10.1016/j.neuroimage.2013.02.008.

[3] “Chapter 8 - The brain is conscious,” in *Fundamentals of Cognitive Neuroscience*, B. J. Baars and N. M. Gage, Eds., San Diego: Academic Press, 2013, pp. 211–252, ISBN: 978-0-12-415805-4. DOI: https://doi.org/10.1016/B978-0-12-415805-4.00008-4.

[4] Barzegaran, E., Bosse, S., Kohler, P. J., and Norcia, A. M., “EEGSourceSim: A framework for realistic simulation of EEG scalp data using mri-based forward models and biologically plausible signals and noise,” *Journal of neuroscience methods*, vol. 328, no. 108377, 2019. DOI: 10.1016/j.jneumeth.2019.108377.

[5] Bonizzi, P., Karel, J. M., Meste, O., and Peeters, R. L. M., “Singular spectrum decomposition: A new method for time series decomposition,” *Advances in Adaptive Data Analysis*, vol. 06, no. 04, p. 1 450 011, 2014. DOI: 10.1142/s1793536914500113.

[6] Breiman, L., “Random Forests,” *Machine Learning*, vol. 45, pp. 5–32, 2001. DOI: 10.1023/A:1010933404324.

[7] Bruña, R., Maestú, F., and Pereda, E., “Phase locking value revisited: Teaching new tricks to an old dog,” *Journal of Neural Engineering*, vol. 15, no. 5, p. 056 011, 2018. DOI: 10.1088/1741-2552/aacfe4.

[8] Carter, G., Knapp, C., and Nuttall, A., “Estimation of the magnitude-squared coherence function via overlapped fast Fourier transform processing,” *IEEE Transactions on Audio and Electroacoustics*, vol. 21, no. 4, pp. 337–344, 1973. DOI: 10.1109/TAU.1973.1162496.

[9] Carter, G. C., “Coherence and Time Delay Estimation,” *Proceedings of the IEEE*, vol. 75, no. 2, pp. 236–255, 1987. DOI: 10.1109/proc.1987.13723.

[10] Cooley, J. W. and Tukey, J. W., “An Algorithm for the Machine Calculation of Complex Fourier Series,” *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965. DOI: 10.1090/s0025-5718-1965-0178586-1.

[11] Couronné, R., Probst, P., and Boulesteix, A.-L., “Random forest versus logistic regression: A large-scale benchmark experiment,” *BMC Bioinformatics*, vol. 19, no. 270, 2018. DOI: 10.1186/s12859-018-2264-5.

[12] Elahiana, B., Yeasina, M., Mudigoudarb, B., Whelessb, J. W., and Babajani-Feremi, A., “Identifying seizure onset zone from electrocorticographic recordings: A machine learning approach based on phase locking value,” *Seizure*, vol. 51, pp. 35–42, 2017. DOI: 10.1016/j.seizure.2017.07.010.

[13] Gevins, A., Smith, M. E., McEvoy, L. K., Leong, H., and Le, J., "Electroencephalographic imaging of higher brain function," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 354, no. 1387, A. Howseman and S. Zeki, Eds., pp. 1125–1134, 1999. DOI: 10.1098/rstb.1999.0468.

[14] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning.* MIT Press, 2016, http://www.deeplearningbook.org.

[15] Guevara, M. A. and Corsi-Cabrera, M., "EEG coherence or EEG correlation?" *International Journal of Psychophysiology*, vol. 23, no. 3, pp. 145–153, 1996. DOI: 10.1016/S0167-8760(96)00038-4.

[16] Hastie, T., Tibshirani, R., and Friedman, J., *The elements of statistical learning: data mining, inference and prediction*, 2nd ed. Springer, 2009.

[17] Hoekstra, R., Kiers, H. A. L., and Johnson, A., "Are assumptions of well-known statistical techniques checked, and why (not)?" *Frontiers in psychology*, vol. 3, no. 137, 2012. DOI: 10.3389/fpsyg.2012.00137.

[18] Hülsemann, M. J., Naumann, E., and Rasch, B., "Quantification of Phase-Amplitude Coupling in Neuronal Oscillations: Comparison of Phase-Locking Value, Mean Vector Length, Modulation Index, and Generalized-Linear-Modeling-Cross-Frequency-Coupling," *Frontiers in Neuroscience*, vol. 13, 2019. DOI: 10.3389/fnins.2019.00573.

[19] James, G., Witten, D., Hastie, T., and Tibshirani, R., *An introduction to statistical learning: With applications in R*, ser. Springer texts in statistics. New York: Springer, 2013, vol. 103, ISBN: 9781461471370.

[20] Lachaux, J.-P., Rodriguez, E., Martinerie, J., and Varela, F. J., "Measuring phase synchrony in brain signals," *Human Brain Mapping*, vol. 8, no. 4, pp. 194–208, 1999. DOI: 10.1002/(sici)1097-0193(1999)8:4<194::aid-hbm4>3.0.co;2-c.

[21] Light, G. A. *et al.*, "Electroencephalography (EEG) and Event-Related Potentials (ERPs) with Human Participants," *Current Protocols in Neuroscience*, vol. 52, no. 1, 2010. DOI: 10.1002/0471142301.ns0625s52.

[22] Lindgren, G., Rootzén, H., and Sandsten, M., *Stationary Stochastic Processes for Scientists and Engineers.* Boca Raton: CRC Press Taylot & Francis Group, 2014, ISBN: 978-1-4665-8618-5.

[23] Lowet, E., Roberts, M. J., Bonizzi, P., Karel, J., and De Weerd, P., "Quantifying Neural Oscillatory Synchronization: A Comparison between Spectral Coherence and Phase-Locking Value Approaches," *PLOS ONE*, vol. 11, no. 1, e0146443, 2016. DOI: 10.1371/journal.pone.0146443.

[24] Nolte, G., Bai, O., Wheaton, L., Mari, Z., Vorbach, S., and Hallett, M., "Identifying true brain interaction from EEG data using the imaginary part of coherency," *Clinical Neurophysiology*, vol. 115, no. 10, pp. 2292–2307, 2004. DOI: 10.1016/j.clinph.2004.04.029.

[25] Ohlsson Mattiasand Edén, P., *Lecture Notes on Introduction to Artificial Neural Networks and Deep Learning (FYTN14/EXTQ40/NTF005F)*, 2016.

[26] Palva, J. M. *et al.*, “Ghost interactions in MEG/EEG source space: A note of caution on inter-areal coupling measures,” *NeuroImage*, vol. 173, pp. 632–643, 2018. DOI: 10.1016/j.neuroimage.2018.02.032.

[27] Persson, S., *An ERP Study of Competitive Memory Retrieval in a Lateralized Auditory Task*, eng, Master Thesis, 2020.

[28] Purves, D. *et al.*, *Neuroscience*, 3rd. Sunderland, MA: Sinauer Associates, 2004, ISBN: 0-13-754920-2.

[29] Reis, P., Hebenstreit, F., Gabsteiger, F., Tscharner, V. von, and Lochmann, M., “Methodological aspects of EEG and body dynamics measurements during motion,” *Frontiers in Human Neuroscience*, vol. 8, 2014, ISSN: 1662-5161. DOI: 10.3389/fnhum.2014.00156.

[30] Sandrone, S. *et al.*, “Weighing brain activity with the balance: Angelo Mosso’s original manuscripts come to light,” *Brain*, vol. 137, no. 2, pp. 621–633, May 2013, ISSN: 0006-8950. DOI: 10.1093/brain/awt091.

[31] Sandsten, M., “Cross-spectrum and coherence function estimation using time-delayed Thomson multitapers,” Jun. 2011, pp. 4240–4243. DOI: 10.1109/ICASSP.2011.5947289.

[32] Srinivasan, R., Winter, W. R., Ding, J., and Nunez, P. L., “EEG and MEG coherence: Measures of functional connectivity at distinct spatial scales of neocortical dynamics,” *Journal of neuroscience methods*, vol. 166, no. 1, pp. 41–52, 2007. DOI: 10.1016/j.jneumeth.2007.06.026.

[33] Stam, C. J., Nolte, G., and Daffertshofer, A., “Phase lag index: Assessment of functional connectivity from multi channel EEG and MEG with diminished bias from common sources,” *Human Brain Mapping*, vol. 28, no. 11, pp. 1178–1193, 2007. DOI: 10.1002/hbm.20346.

[34] Taha, A. and Hanbury, A., “Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool,” *BMC Med Imaging*, vol. 15, no. 29, 2015. DOI: 10.1186/s12880-015-0068-.

[35] “The Theta-Gamma Neural Code,” *Neuron*, vol. 77, no. 6, pp. 1002–1016, 2013, ISSN: 0896-6273. DOI: 10.1016/j.neuron.2013.03.007.

[36] Vinck, M., Oostenveld, R., Wingerden, M. van, Battaglia, F., and Pennartz, C. M., “An improved index of phase-synchronization for electrophysiological data in the presence of volume-conduction, noise and sample-size bias,” *NeuroImage*, vol. 55, no. 4, pp. 1548–1565, 2011. DOI: 10.1016/j.neuroimage.2011.01.055.

[37] Welch, P., “The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms,” *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967. DOI: 10.1109/TAU.1967.1161901.

[38] Wheaton, L. A., Nolte, G., Bohlhalter, S., Fridman, E., and Hallett, M., “Synchronization of parietal and premotor areas during preparation and execution of praxis hand movements,” *Clinical Neurophysiology*, vol. 116, no. 6, pp. 1382–1390, 2005. DOI: 10.1016/j.clinph.2005.01.008.

A Appendix

Simulated data

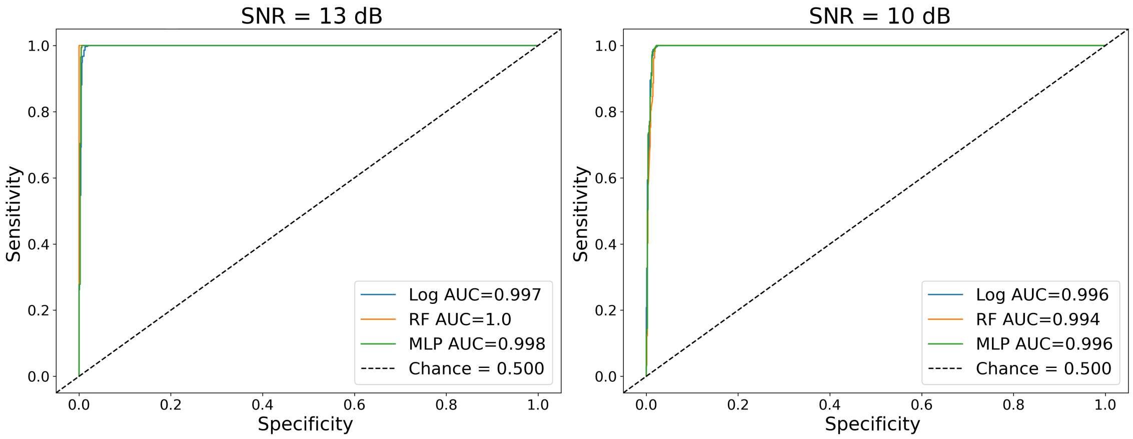


Figure 18: ROC curve for the SigDet based models for different SNR.

Real Data

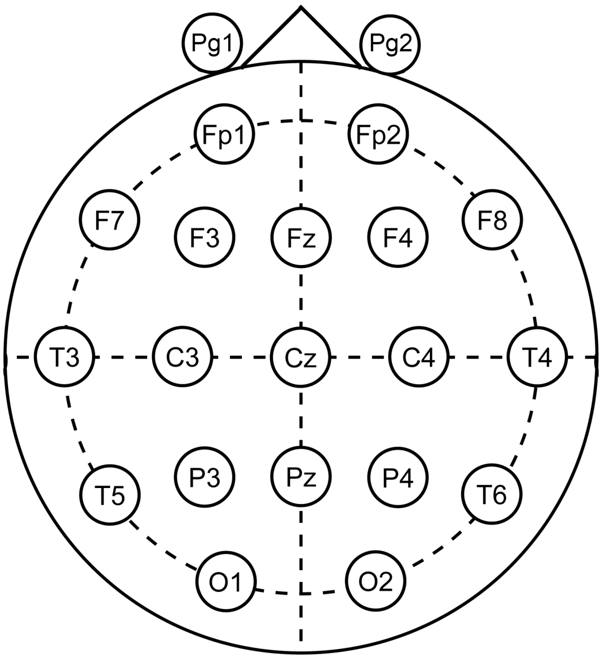
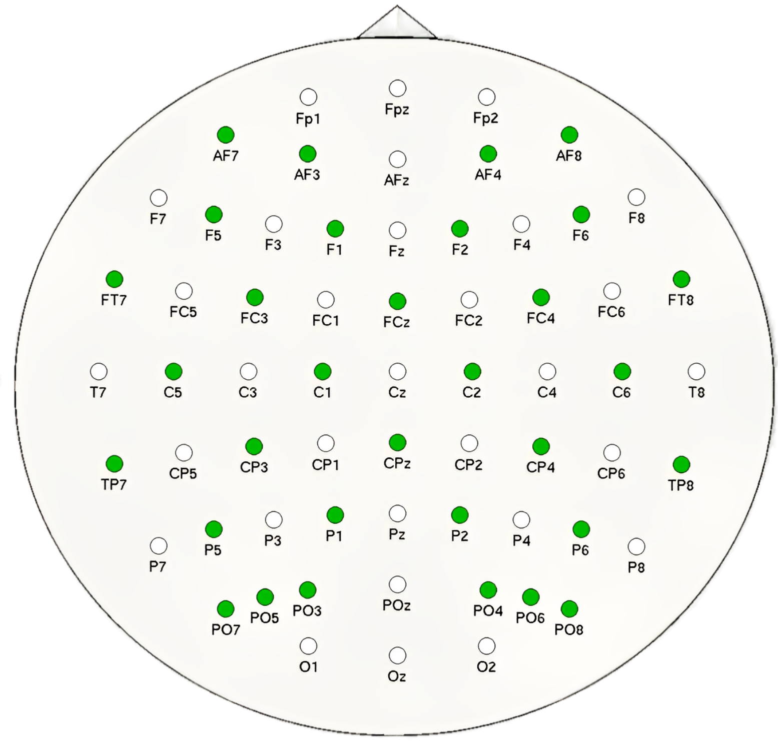


Figure 19: Placement of electrodes used to record the visual EEG data in Section 4.1



Electrode pairs
T7-T8
TP7-TP8
C5-C6
FT7-FT8
P7-P8
CP5-CP6
C3-C4
F7-F8
Fz-Pz
F3-P4
P3-F4
FC3-CP4
CP3-FC4
F1-P2
P1-F2

Figure 20: Placement of electrodes used to record the auditory data in Section 4.2 (left) as well as the chosen electrode pairs (right). The pairs marked in bold are used for both sets of models.

Table 7: The final choice of hyperparameters for the two sets of models trained on real EEG data from Section 4.2, where $\boldsymbol{N}$ is the total number of features.

Set of models	**Subject**	**Hyperparameter**							$\boldsymbol{N}$
		B	m	S	H	$\varphi(a)$	Opt	r	
Side	11	5	2	8	5	tanh	sgd	10^{-6}	10
Significance	13	10	2	256	10	relu	ADAM	10^{-4}	8
		RF paramaters			ANN paramaters				

Master’s Theses in Mathematical Sciences 2022:E78
ISSN 1404-6342

LUNFMS-3117-2022

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lu.se/