UTRECHT UNIVERSITY
Faculty of Science
Artificial Intelligence
Master's thesis

Foreseeing Electrical Activity of the Brain – Generative Deep Learning Models for EEG Time Series Forward Prediction

Hanna Pankka

July 20, 2020

Utrecht University

Supervisor
Dr. Timo Roine
First examiner
Prof. Leon Kenemans
Second examiner
Dr. Ben Harvey

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Timo Roine for providing me with excellent guidance throughout this process as well as for always taking the time to help and answer my questions. I would also like to thank prof. Leon Kenemans and Dr. Ben Harvey for acting as examiners for this thesis.

I am grateful for prof. Risto Ilmoniemi for the great opportunity to carry out this work at ConnectToBrain and for suggesting this research topic. Thanks also to Riku, Olli-Pekka, Johanna, Roberto, Tuomas, Pantelis, Aino, Mikko, and everyone else in C2B for discussions, comments, and for showing interest in my work.

Thanks also to my friends and family for your love and support. An especially warm thanks to my beloved fiancé Anttoni: thank you for your endless support during this whole process as well as for spending a fortune on flights to the Netherlands.

Lastly, I would like to honor the memory of prof. Timo Honkela, without whom I probably would not be on this path.

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 810377).

Abstract

Recent findings suggest that efficacy of transcranial magnetic stimulation (TMS) can be substantially improved with brain-state dependent stimulation. This can be done with a brain-computer interface (BCI) that triggers the stimulation based on real-time measured EEG. However, this is challenging, as algorithmic decision making takes time and brain states are known to change rapidly. One solution here is to forward predict the EEG time series – this enables the BCI to anticipate the occurence of brain states that are suitable for stimulation.

In this thesis we propose two convolutional neural network models for forecasting EEG time series. The first one is an adaptation of the WaveNet model developed for processing audio signals. The second one in turn is a multivariate adaptation of the first one.

We found that our univariate model is better at estimating instantaneous phase of an EEG signal compared to an autoregressive forward prediction model that has been previously used for brain-state dependent TMS. In addition, our multivariate model was not able to achieve more accurate predictions than our univariate model, but it did show slightly improved phase estimation accuracy.

In conclusion, results reported here indicate that deep learning is a feasible approach for EEG time series forward prediction.

Contents

1 Introduction

The idea of using pure thought to communicate with computers has been exciting humans for a long time. Thus, for already a few decades the field of brain-computer interfaces (BCI) has been gaining interest and new applications. A BCI refers to an electrode-computer construct, where inputs from brain recording electrodes are transformed into functional outputs by a computer (Krucoff, Rahimpour, Slutzky, Edgerton, & Turner, 2016). They have been developed for many purposes, especially in health care, where both assistive and rehabilitative BCIs have been introduced (Krucoff et al., 2016). Examples include BCIs for moving prosthetic fingers (Hotson et al., 2016) and controlling paralysed muscles (Moritz, Perlmutter, & Fetz, 2008) as well as facilitating motor recovery from stroke (Buch et al., 2008; Gharabaghi et al., 2014) and paraplegia (Donati et al., 2016).

The rise of BCI applications has been creating demand for new technologies in cross section of artificial intelligence and neuroscience. A key part of all BCIs is algorithmic solutions needed for translating raw inputs from the nervous system into desired actions. This can include tasks such as recognising particular brain states and anticipating how the system develops.

An interesting area in the field of BCIs is brain state dependent neuromodulation, where a brain stimulation device is controlled through a BCI that processes neuronal activity in real time and detects brain states that are suitable for stimulation. One widespread neurostimulation method is transcranial magnetic stimulation (TMS) – a non-invasive tool that can excite or inhibit neuronal activity with changing magnetic fields. This thesis is carried out as a part of ConnectToBrain-project that aims at developing a closed-loop multi-locus transcranial magnetic stimulation (mTMS) device for neurorehabilitation. The planned mTMS device would adjust stimulation parameters, such as timing and location, according to ongoing brain states and gathered feedback on the efficacy of the stimulation.

One of the key ideas here is that when one is in hopes of doing successful manipulation of any system, be it a political environment, a pool of microbes or a pot on a stove, one needs to know that particular system: how it works, how to initiate change within that system and what the system looks like at time of intervention. In this regard the brain is no different from any other system. By knowing the current state of the brain at each moment and how possible actions influence that particular state, we can have substantially more control over the stimulation outcomes – and, ultimately, achieve more effective stimulation in comparison to current practices.

In the case of the brain, factoring in the current state of the system is extremely difficult, as drastic changes in activity can take place within milliseconds. This poses a fundamental challenge

for precisely timing the stimulation as algorithmic decision making takes time. For this reason, one essential building block of BCI controlled TMS is foreseeing how brain states are evolving: if the goal is to target stimulation at certain brain states, we need to know in advance when said brain states are going to occur – otherwise, the pulses will inevitably arrive later than intended. The more precisely we can estimate the prognosis of EEG time series the better we can target the stimulation.

EEG time series forward prediction has already been of interest in research for decades (see for example Blinowska & Malinowski, 1991; Hernández, Valdés, Biscay, Jiménez, & Valdés, 1995). To this date the forecast methods have mainly consisted of relatively simple linear and non-linear methods like the autoregressive model (AR) including its variants such as autoregressive moving average model (ARMA) and autoregressive integrated moving average model (ARIMA). However, recent advancements in machine learning techniques, in deep learning in particular, and their successful application in EEG classification problems as well as in time series analysis in other fields, such as economics (see for example Zhou et al., 2016), suggest that analysis of neural time series could also benefit from these tools.

In this thesis we will propose a convolutional neural network (CNN) model for forecasting resting state EEG time series. This model uses a fragment of EEG signal to predict the subsequent fragment of the same signal. Performance of this model will then be compared to the current state-of-the-art EEG forward prediction method in real time brain state dependent stimulation – the AR model (Zrenner, Desideri, Belardinelli, & Ziemann, 2018). The aim here is to outperform the AR model in predicting the upcoming value sequence of one EEG channel in an offline analysis.

In addition, we attempt to further improve the predictions of our CNN model by presenting a multivariate extension of it. The idea here is that providing information from also other channels would improve the predictions due to signal propagation within the brain. In other words, we hypothesize that adding data from concurrent channels can help making better predictions because those signals carry additional information about the signal we are trying to predict. To our knowledge, this has not been attempted before as the models used for predicting EEG time series have all been univariate models, i.e. they have utilized data from only one channel.

The rest of this thesis consists of four parts. In the next section, we will explain the key concepts and provide a summary of previous work done in this field. In sections three and four we will explain the methodology of this study and present the results. Finally, the fifth section will consist of a discussion of the results as well as conclusions of this study.

2 Background

2.1 TMS and Brain-State Dependent Stimulation

Transcranial magnetic stimulation (TMS) is a non-invasive tool for brain research and clinical use. By using changing magnetic fields, TMS can excite or inhibit neural activity. Due to its capability of producing effects in the brain that last beyond the stimulation it is increasingly used as a treatment for different psychiatric and neurological disorders such as stroke, depression and chronic pain (see for example Hallett, 2000; Lefaucheur et al., 2020). However, the treatment responses vary largely between individuals and effect sizes have been modest so far (Lefaucheur et al., 2014). Possible big contributors to this are varying treatment practises, large inter-individual variability and variation introduced by ongoing brain states.

Traditionally TMS has been applied manually to one location at a time and without knowledge on the ongoing brain oscillations during the stimulation. Lately, there has been growing evidence showing that synchronizing the TMS pulses with relevant ongoing brain oscillations can improve its efficacy (see for example Gharabaghi et al., 2014; Zrenner et al., 2018; Kraus et al., 2016). For this reason, the outcomes can be better controlled with brain-state dependent stimulation: plasticity induction can be made more effective for example with activity-dependent stimulation that invoke Hebbian strenghtening of neural connections (see for example Gharabaghi et al., 2014) or by targeting the negative peaks of an oscillation (see for example Zrenner et al., 2018).

So far, the applications of brain-state dependent TMS have all been open-loop systems, where the stimulation parameters have been tuned prior to the stimulation session. The ConnectToBrain-project aims at improving on this by taking the brain-state dependent stimulation a step further with closed-loop EEG-mTMS. By observing in real time the effects of the stimulation, the stimulation parameters can be algorithmically tuned online. With suitable, real-time controlled and individualized stimulation parameters, including for instance timing, locus, and intensity of the stimulation, we can optimize the stimulation effects.

2.2 Electroencephalography

Electroencephalography (EEG) is a non-invasive method for measuring the electrical activity of the brain. It is a century old technology that has been widely used in research areas such as neuroscience and psychology and for clinical diagnosis purposes already for decades. Lately, it has also been navigating its way into a whole new rising domain of technology, namely, to the world of

brain-computer interfaces and brain-state dependent neuromodulation (Biasiucci, Franceschiello, & Murray, 2019).

As said, EEG is a method for measuring electric fields in the brain. What this means is that the EEG device measures electric potential differences, voltages, of postsynaptic potentials taking place in the brain. By postsynaptic potentials we refer to inhibitory and excitatory neurotransmitter releases in synapses (Biasiucci et al., 2019).

An EEG device measures electric fields induced by these postsynaptic potentials with electrodes placed on the scalp. The amount of electrodes varies between devices – while the most conventional systems have around twenty electrodes, the various implementations of the EEG device range from a simple system with only two electrodes to high-resolution systems with 60 or up to 128 electrodes. During recording we receive a signal from each channel. These signals, that together constitute the EEG recording, represent magnitudes of voltage as a function of time. EEG thus allows real time tracking of electrical activity within the cortex (Biasiucci et al., 2019).

For analysis purposes the EEG signal is often band-pass filtered into frequency bands. A common way to divide these frequency bands is to divide them into delta, theta, alpha, beta and gamma waves (δ=0.2–3.5, θ=4–7.5, α=8–13, β=14–30 and γ=30–90 Hz) as well as into high frequencies (> 90 Hz) (Biasiucci et al., 2019).

One clear advantage that EEG has over many other brain research methods is that it provides highly accurate time resolution (Biasiucci et al., 2019). For this reason, EEG is an excellent tool for when we are interested in immediate responses to presented stimuli or in real time monitoring of the brain, which is a crucial component of BCIs. Other advantages of EEG include its relatively cheap price that makes it accessible and its suitability to be used for measuring signals from people of all ages, starting from new-borns, and even from animals. Due to its noninvasiveness, it is also really convenient and safe to use. In addition, EEG can be used simultaneously with other brain mapping and imaging tools, such as MRI and TMS, which is a requirement for an online monitoring device that is to be used in brain-state dependent neurostimulation (Biasiucci et al., 2019).

Despite its convenientness in a wide range of applications and its firm foothold in various fields there are also some fundamental challenges and difficulties with EEG data and their recording and analysis. First of all, EEG is able to measure only a portion of all the electrical activity going on in the cortex. Furthermore, as the electrodes are placed on the scalp, nearby signals are mixed together and damped by the skull. In addition, EEG data are non-stationary and generally extremely noisy as well as heavily prone to gathering artifacts from e.g. eye movements and environment in which

the measurement is performed. EEG data also have relatively poor spatial resolution compared to some other brain mapping tools such as MEG and MRI and it has large inter-subject variability. Lastly, EEG data are high-dimensional which makes it a difficult target for analysis and especially for forward prediction of the time series.

In the next subsection I will explain basic concepts of machine learning and artificial neural networks, after which I will proceed in reviewing studies that have been using deep learning and time series analysis methods for processing EEG data.

2.3 Machine Learning

Machine learning has been one of the most trending fields of this century. Although researchers have been playing with the idea of thinking and learning machines at least since the 1950s, most of the major advancements have been happening during the last decades due to the increasing amount of accessible data and processing resources.

Common machine learning methods include for example decision trees, logistic regression and artificial neural networks. As the focus of this thesis is in artificial neural networks (ANN) my main focus in this chapter is in ANNs; however, most of the principles described here apply to all machine learning methods.

One of the most well-known definitions of machine learning was presented by Tom Mitchell in 1997:

> "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." (Mitchell, 1997).

A key feature in machine learning thus is that a programmer does not explicitly tell the computer how to perform a given task but rather the computer finds a way to do it on its own. However, the programmer does set some boundary conditions about how the learning happens. One of these predefinitions is the type of learning. These different techniques are supervised, unsupervised, semisupervised and reinforcement learning (Mohammed, Khan, & Bashier, 2016).

Supervised and unsupervised learning differ from each other in regards to how the algorithm is guided during the learning process. In supervised learning the data set always contains labels that explain the input data; so for every input X there exists a label that describes X (Mohammed et al., 2016). As the algorithm then has a pile of examples with their correct descriptions, it

can learn the underlying characteristics of the examples with respect to their labels. Instead, in unsupervised learning, as the name suggests, there are no 'correct answers' available; hence the training process cannot be guided by the labels, so the results arise from the training examples themselves. Semisupervised learning, in turn, is a mixture of supervised and unsupervised learning, with both labeled and unlabeled training examples.

Machine learning algorithms have been developed to solve multiple types of tasks. The main three tasks are classification, regression and clustering, of which the two former are examples of supervised learning, whereas the last one represents unsupervised learning. The supervised learning tasks, classification and regression, focus on mapping inputs to correct labels: In classification tasks the goal is to categorise the inputs into predefined classes, such as 'dogs' and 'cats'. Regression tasks, in turn, are about mapping each input to a numerical value; regression tasks, thus, are for describing measurable things such as 'weight'. The unsupervised learning example, clustering, aims at fractioning the input data into clusters according to the properties of the data; and as there are no labels available in unsupervised learning, the clustering is done purely based on the similarities of the data characteristics found amongst the training examples.

As mentioned in the beginning of this section, one of the common machine learning methods are artificial neural networks. ANNs are an architecture inspired by the way the biological brain treats information (Haykin, 2010). A simple feedforward ANN architecture consists of an input layer, possibly one or more hidden layers and an output layer. The layers in an feedforward ANN consist of one or more nodes ('neurons') and the layers project directly onto the following layer (Haykin, 2010). If the network is said to be fully connected all the nodes in every layer are connected to all of the nodes in the following layer. Each of these connections has a weight that is initialised to a random number and the resulting outputs depend on these weights. During the training phase of a supervised ANN the output after each run is evaluated against the correct answer, i.e. the label: this constitutes the *error signal*. The algorithm then tweaks the connection weights to minimize the error, which is evaluated with a performance measure, such as mean absolute error or sum of squared errors.

2.3.1 Deep Learning

Deep learning is a form of machine learning that has lead to breakthroughs in many computational fields where computers previously have not been succeeding (LeCun, Bengio, & Hinton, 2015). These include tasks such as image, video, and audio processing.

For many simpler machine learning methods it is necessary to design a set of features that the model can then utilize for learning the correct interpretations of its inputs. Designing such sets of features by hand can however be difficult and time consuming. Deep learning is a form of representation learning where the model also learns the *features themselves* and not just the mapping from the features to outputs. This is made possible by building complicated concepts out of simpler ones, essentially forming a deep hierarchy of concepts; hence also the name (Goodfellow, Bengio, & Courville, 2016, chapter 1).

Two of the most widely used deep learning architectures are recurrent and convolutional neural networks (RNN and CNN, respectively). RNNs are deep learning architectures developed specially for treating sequential data. RNNs make use of the idea of parameter sharing: each part of the output is a function of all the preceding parts and all of the parts are produced using the same update rule (Goodfellow et al., 2016, chapter 10). This is how RNNs are able to learn long chains of dependencies in the input data. RNNs have performed particularly well in tasks such as natural language processing.

CNNs in turn have been excelling in tasks involving grid-like data structures, such as images and time series data that can be thought of as 2D and 1D grids, respectively (Goodfellow et al., 2016, chapter 9). The main building block of CNNs is a convolution operation that generally is defined as an operation between two functions that produce a new function (Goodfellow et al., 2016, chapter 9). In the context of CNNs the two functions are usually referred to as input and kernel, while the output of the operation is called a feature map (Goodfellow et al., 2016, chapter 9). Multiple filters are typically used in one convolutional layer, resulting in various feature maps, each describing one feature in the input. Neural networks utilizing convolutional layers have resulted in state-of-the-art performance in for example image recognition.

Next, we will briefly introduce a few key concepts of deep learning that are used in this study.

Activation functions

Activation functions are commonly used after some or all of the layers in a deep neural network. The role of the activation functions is to introduce non-linearity to the network which enables the model to learn more complex relationships (Goodfellow et al., 2016, chapter 6). Below are presented activation functions that will be used in this study.

ReLU

$$f(x) = \begin{cases} x & \text{if } x \geq 0, \\ 0 & \text{otherwise} \end{cases}$$

LeakyReLU

$$f(x) = \begin{cases} x & \text{if } x \geq 0, \\ \alpha \cdot x & \text{otherwise } (\alpha \geq 0) \end{cases}$$

Tanh

$$f(x) = tanh(x)$$

Sigmoid

$$f(x) = \frac{1}{1 + e^{-x}}$$

Convolution operation

In a convolution layer, a filter is slid through the input, producing a filter map as a result (Goodfellow et al., 2016, chapter 9).

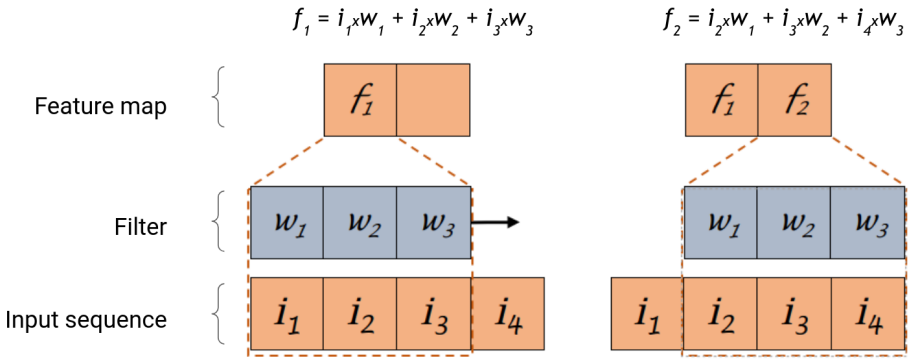


Figure 1: An example of a 1D convolution operation with a filter of length 3. A filter with weights (w_1, w_2, w_3) is slid through an input sequence, producing a feature map (f_1, f_2).

Padding

In a valid convolution operation the kernel is always applied such that it fits entirely within the input (Goodfellow et al., 2016, chapter 9). However, this leaves the edges of the input unprocessed, resulting in the feature map to be smaller than the input. If we want to prevent this, we can add zero padding to the input.

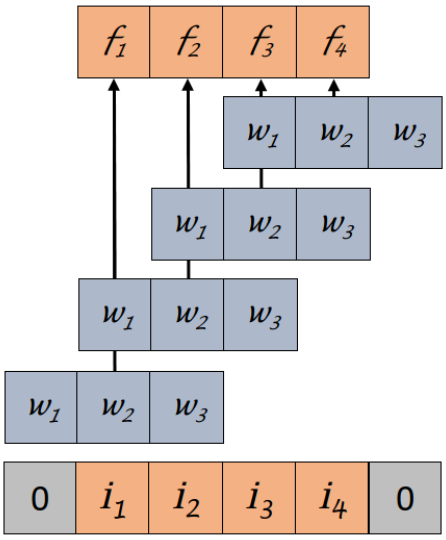


Figure 2: Shape preserving padding.

Softmax

Softmax is a function that is often used as the output layer in a classifier network (Goodfellow et al., 2016, chapter 6). It gives a probability distribution over n possible classes – this distribution can then be used to pick the most likely class of each input.

Kernel initialization

In the beginning of a training process of a neural network model, weights of each layer need to be initialized. Typically, a random set of weights is chosen. The kernel initializer method then determines, *how* the random weights are picked. For this study we use He initializer (He, Zhang, Ren, & Sun, 2015) that is especially suitable to be used with ReLU activations. The He initializer draws the weights from a 0 centred truncated normal distribution.

2.3.2 Deep Learning in EEG Analysis

In section 2.2 we shed light on some built-in difficulties of EEG data processing, such as low signal-to-noise ratio, high-dimensionality and non-stationarity. In the previous section (2.3.1), we saw that deep learning has shown state-of-the-art performance in processing other complex data like images, video, text and speech (LeCun et al., 2015). These promising results from other fields give reason to presume that deep learning would be able to overcome some of the challenges also in EEG signal processing and thus lead to good results in this field.

Thus, it is of no surprise that numerous studies at increasing frequency have already been published on using deep learning for EEG. Majority of these experiments are classification studies and the most frequently used model architectures are convolutional neural networks; other popular architectures include recurrent neural networks and autoencoders (Roy et al., 2019). Popular research topics in this domain include for example emotion recognition, motor imagery (Craik, He, & Contreras-Vidal, 2019) and EEG feature learning (Roy et al., 2019). Furthermore, deep learning has also been succesfully applied for predicting event-related potentials from EEG time series (Ibagon, Kothe, Bidgely-Shamlo, & Mullen, 2018).

Deep learning is argued to have several advantages over other methods in EEG processing. Firstly, as deep neural networks are excellent at extracting features, they can be used to analyse EEG data that are only minimally, if at all, preprocessed. The features deep neural networks extract might also be more expressive compared to those hand designed by humans (Roy et al., 2019). This also makes the data processing less dependent on specific domain knowledge about the data and extensive work of trained professionals (Craik et al., 2019). In addition, the usage of deep learning enables advancing areas of analysis, such as generative modeling and domain adaptation, that are tricky to do with other tools (Roy et al., 2019).

However, there are also challenges in applying deep learning for EEG analysis. One of the most prevalent of these is the lack of suitable data sets (Roy et al., 2019); training deep learning models requires vast amounts of data and EEG data are not as massively available as for example image data are. Secondly, EEG data is different from the types of data that deep learning is usually used for; it has been argued that for this reason it might be that many deep learning methods are not applicable as such to EEG data (Roy et al., 2019).

2.4 Previous Approaches in Forecasting Time Series

A time series is defined by Box et al. as "a sequence of observations taken sequentially in time" (Box, Jenkins, Reinsel, & Ljung, 2008, p. 1). Time series data are used in various fields such as meteorology, economics, engineering and social sciences. This kind of data are typically discrete and gathered e.g. monthly, weekly or hourly. In regards to this thesis, of particular interest in time series data is that future observations are often dependent on one or more of the previous observations. If there exists a causal relationship of some form between the time series' values, it is justifiable to assume that future values can be predicted, at least to some extent, based on previous values in the series (Box et al., 2008).

Numerous models have been developed for this forecasting purpose. These models can be categorised into linear and non-linear models and further to univariate and multivariate as well as stationary and nonstationary models. Examples include models such as autoregressive (AR), non-linear autoregressive (NAR) and vector autoregressive model (VAR).

Furthermore, diverse types of artificial neural networks have also been implemented. Examples include tasks such as forecasting traffic flow (Ren, Wang, Yin, Chen, & Shan, 2013), financial time series (Borovykh, Bohte, & Oosterlee, 2017; Pulido, Melin, & Castillo, 2014; Zhou et al., 2016), weather (Zaytar & El Amrani, 2016), and wind speed (Doucoure, Agbossou, & Cardenas, 2016). In all of these cases, artificial neural networks have proved to be beneficial for the modeling problem at hand.

In the next subsection we will give a brief overview on methods previously used for forward predicting EEG time series.

2.4.1 EEG Forward Prediction

Both linear and non-linear univariate models have been implemented for predicting EEG time series. One of the most frequently used linear univariate model for this purpose is autoregressive forward prediction model (AR). An autoregressive process estimates the next value in a sequence based on the previous values:

$$\tilde{z}_t = a_t + \sum_{j=1}^{p} \phi_j \tilde{z}_{t-j} \ ,$$

where $\{\tilde{z}_{t-1}, \tilde{z}_{t-2}, \ldots, \tilde{z}_{t-p}\}$ are the past p values, $\phi_1, \phi_2, \ldots, \phi_p$ represent the weights of the past values and a_t is white noise. (Box et al., 2008, pp. 47–53)

The usage of the AR model in the context of EEG time series is not new. For example, already in 1991 Blinowska and Malinowski published a study where they compared the AR model with a non-linear univariate prediction model (Blinowska & Malinowski, 1991). The model in question, originally proposed by Sugihara and May (1990), considers first N points from an array $X_t = \{x_t, x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(E-1)\tau}\}$ (with E and τ being an embedding dimension and a lag, respectively) to predict the following p steps. The formula for this model is the following:

$$x_{j+p} = \sum_{i=1}^{E+1} \tilde{x}_{k_i+p} exp[-\alpha dist(x_j, x_{k_i})] \ ,$$

where $j > N, k + p < N$ and $\tilde{x}_{k_i}$ denotes the closest neighbours of x, α is a constant and $dist$ is Euclidean distance in E dimensions (Blinowska & Malinowski, 1991, pp. 159–160). The most noteworthy difference compared to the linear AR model is that here the weights are exponential and the input sequence is not composed of consecutive values of the observed value sequence.

For the comparison of the linear and non-linear models a 5th order AR model was used. The non-linear model had an embedding dimension E of 2 or 3, depending on the channel that was predicted. The lag τ was set equal to sampling interval and $\alpha = 0.005$. By comparing these two models, they found that the prediction difference was not too large but that the linear AR model perfomed slightly better in all comparison conditions (Blinowska & Malinowski, 1991).

In addition, a few approaches utilizing artificial neural networks have been proposed for EEG time series prediction. In 2011, Samanta examined the use of single multiplicative neuron (SMN) and adaptive neurofuzzy inference system (ANFIS) for predicting chaotic time series (Samanta, 2011). In the forecast task on test set, the SMN model performed modestly (NRMSE = 0.5618) whereas the ANFIS model's predictions were considerably better (NRMSE = 0.2189). Later, Kose and Arslan (2017) introduced a model that combines the ANFIS model with vortex optimization algorithm (VOA). This model yielded promising results: the model was tested on four data samples and it performed quite well on all of those, so the authors argue that the ANFIS-VOA model is generally able to predict the prognosis of EEG (Kose & Arslan, 2017).

Although these previous attempts in modeling the EEG time series have had satisfactory results, there is still room for supplemental approaches. Because of the challenges of analysing EEG data – high dimensionality, low signal-to-noise-ratio – (see section 2.2) and because of the successful application of deep learning methods on similar problems in other fields as well as in EEG classification problems, we have a reason to believe that EEG time series prediction would also benefit of these methods.

In the next section we will introduce deep learning methods that have been previously used in similar problems.

2.4.2 Recurrent and Convolutional Neural Networks

In addition to treating the EEG time series forecasting as a strictly time series modeling problem, we can also have a more general take on the issue. As said, time series is essentially a sequence of values. Consequently, time series forward prediction can also be thought as a subclass of the domain of sequence-to-sequence modeling (S2S), where the goal is to map values of one sequence into another sequence. For this reason, when deciding on the best ways to treat the forward prediction problem it is relevant to consider also models that have not been developed specifically for time series analysis but to deal with other kinds of S2S problems. These include models such as recurrent neural networks and one dimensional convolutional neural networks.

S2S deep learning methods have been extensively studied during the last decade. Most of these models have been developed for natural language processing purposes: a standard example is machine translation where a sequence of words in one language needs to be mapped into a sequence of words in another language in such a way that these two sequences carry the same meaning. Significant progress in this domain has been achieved during the last years.

Perhaps the most well-known group of deep neural networks for S2S tasks are recurrent neural networks (RNN) that are specifically designed to process sequential data. RNNs are essentially computational graphs, where outputs are defined as functions of the previous members (Goodfellow et al., 2016, chapter 10). This way these networks can form recurrent connections that are particularly suitable for modeling sequential data. Another prominent deep learning approach for sequential data are convolutional neural networks (CNN). Although originally developed for images, CNNs have increasingly been adopted in the domain of time series analysis as well (see for example Cui, Chen, & Chen, 2016; Liu, Hsaio, & Tu, 2018).

Despite RNNs being designed for sequential data and CNNs not, recently many have argued that in fact a lot of the time CNNs actually are more suitable for this (see for example Bai, Kolter, & Koltun, 2018). One major pitfall of RNNs is that they can be computationally really expensive especially for long sequences as the piling recurrent connections can quickly make the memory load extremely heavy (see for example Bai et al., 2018; van den Oord, Dieleman, et al., 2016). This can also lead to unnecessarily long training time. As CNNs do not use recurrent connections, they can get off with lower memory requirements and faster training times compared to RNNs (Bai et al.,

2018; van den Oord, Dieleman, et al., 2016). Additional advantages of CNNs over RNNs include possibility for processing the convolution operations in parallel as well as the ability of CNNs to escape the vanishing gradient problem that has been a fundamental challenge for RNNs (Bai et al., 2018). For these reasons, the usage of CNNs for sequence modeling has been rising recently.

Additionally, various CNN based multivariate methods, where the input consists of multiple time series at a time, have been suggested for time series analysis. Some of these are created for classifying time series (see for example Cui et al., 2016; Liu et al., 2018; Yang, Nguyen, San, Li, & Krishnaswamy, 2015) whereas others tackle regression problems (see for example Babu, Zhao, & Li, 2016; Borovykh et al., 2017). Based on the results from these models it seems that for also multivariate time series analysis CNNs are a great choice. Stojov et al. (2018) note that this is because multivariate time series data are natural to display as a space-time image where time series of fixed length are piled up – and as mentioned previously, CNNs are particularly good at interpreting images (Stojov, Koteli, Lameski, & Zdravevski, 2018).

The next two subsections will cover in detail two time series models that are going to be used in the present study: the autoregressive forward prediction model that will be used as a comparison in this study and the WaveNet model that the models we are going to present are based on.

2.4.3 Autoregressive Model in Real Time Phase Estimation

The autoregressive forward prediction model (AR) is the current state-of-the-art model in real time EEG forward prediction for brain state dependent stimulation. In this section we will explain how this model was put to use in an excitability study by Zrenner et al. (2018).

Zrenner et al. used the AR model in a study where they explored how stimulation efficacy is affected by the phase of the neural oscillation at the time of stimulation. More specifically, the goal of this paper was to study differences between TMS induced excitability when the stimulation is targeted in the positive versus negative peak of the alpha-band (8-12 Hz) µ-oscillations in sensorimotor cortex.

The AR model was used to estimate the instantaneous phase of Hjorth-C3 signal to detect the occurences of the positive and negative peaks. The estimation was performed for a 500 ms sliding window with sample rate of 500 Hz. Firstly, the Hjorth-C3 signal was formed with an orthogonal source derivation. This was done for C3 channel with its surrounding four channels (C1, C5, FC3, CP3): the means from all signals were removed and a 5-point sum-of-difference operation applied. The signal was then band-pass filtered to 8-12 Hz with a two-pass band-pass FIR filter of order 128.

After the preprocessing, 64 ms from both sides of the 500 ms sliding window were removed. The AR parameter estimation with 30 order Yule-Walker method was then done using the remaining 372 ms window. Finally, a prediction was made 128 ms forward. As 64 ms from both sides were removed, the prediction then spans from -64 ms to 64 ms such that "time zero" (or "now") is in the middle of the prediction.

To estimate the phase of the oscillation at "time zero", the predicted signal was Hilbert transformed. The Hilbert transform calculates an analytic signal from the original signal – the phase angel can then be calculated from the transformed analytic signal. Here phase angle of 0° represents the positive and 180° the negative peak. Additionally, they calculated the power spectrum from the entire 500 ms sliding window. TMS was then triggered, if a predefined power threshold was exceeded and, depending on the stimulation condition, either positive or negative peak was detected.

True phases in both positive and negative peak conditions were assessed in an offline analysis. The average precisions of the online phase estimations for both conditions were really good (0° and 181°), but the standard deviations were relatively large: 53° and 55°.

2.4.4 The WaveNet Model

In 2016, van den Oord et al. introduced the WaveNet model for creating raw audio signals. WaveNet is a fully probabilistic generative autoregressive deep learning model that has shown state-of-the-art performance in text-to-speech modeling. In addition, it was succesfully used to generate novel and realistic samples of music.

The WaveNet model applies ideas from previous approaches in modeling complex distributions with neural autoregressive generative models. Examples include both picture (van den Oord, Kalchbrenner, & Kavukcuoglu, 2016) and text (Jozefowicz, Vinyals, Schuster, Shazeer, & Wu, 2016) generation. In Wavenet, the main ingredients are dilated causal convolutions, a softmax layer as well as residual and skip connections. We will next briefly introduce each of these components and their usage in this model.

As audio data contain different kinds of structures at different time-scales it has traditionally been extremely hard to model well. WaveNet, however, is particularly good at modeling long range temporal dependencies. For a deep learning model this is a tricky goal to achieve as modeling long sequencies of data is usually computationally extremely expensive: modeling long term variation with convolution operations requires increasing the receptive field, i.e. the filter – and the larger the filter, the more computations are needed. How WaveNet succeeds in this is by using *dilated*

convolutions. A dilated convolution is a normal convolution operation seasoned with a stretched filter: unlike a normal convolution filter of length L that is applied over an area of length L, a dilated filter is applied over an area of length greater than L (see Figure 3). This is achieved by skipping input values according to the applied dilation rate. This allows us to significanty increase the size of the receptive field without immoderately growing the computational cost.

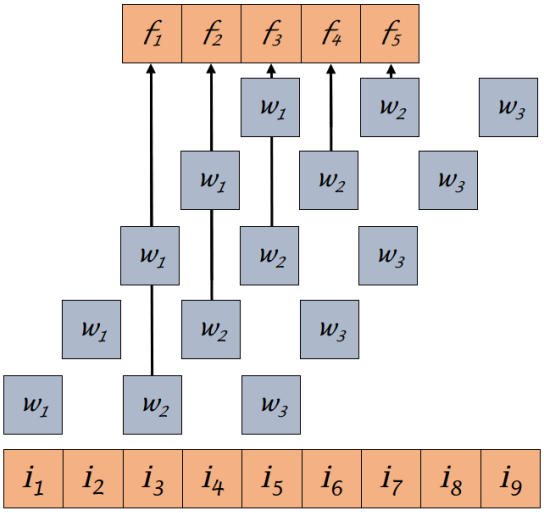


Figure 3: Dilated convolution. A filter of size 3 with dilation rate of 1 spans over 5 input values, whereas the same filter without dilation can only cover 3 values at a time.

The convolutions used here are called causal because the convolution operation is applied in accordance with the ordering of the original data. This is implemented by using causal padding instead of shape preserving padding (see Figure 4). Using causal convolutions is important in modeling time series, because the ordering of the value is of significant importance: future timesteps cannot be included in the estimation of current timestep but the input used must be strictly limited to preceding values only.

WaveNet models each timestamp as a probability distribution, where each value is conditioned on the previous values and once a value is computed it is fed back to the network to take part in predicting the next value. Hence the predicted sequence is a product of conditional probabilities:

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t | x_1, x_2, \ldots, x_{t-1})$$

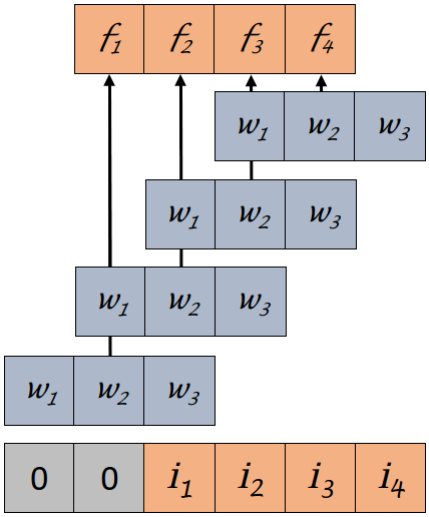


Figure 4: Causal padding. The padding of an input sequence can be made causal by locating the entire padding to the left edge of the sequence (cf. Figure 2). This way each value in the sequence is only dependent on itself and values preceding it.

The output of the model, i.e. the probability distribution of one timestep, is given by a softmax layer. As each timestep in a raw audio signal has 65,536 possible values, computing the probability distribution quickly becomes too heavy to compute. For this reason the possible outputs were quantized to 256 possible output categories. The softmax layer then assigns for each of these 256 categories a probability that represents the likelihood that the next timestep belongs to the respective category.

The model architecture is mainly composed of residual blocks that contain dilated convolution layers, followed by tanh- and σ-activations. The dilation is doubled in each convolutional layer up to 512 (1, 2, 4, ..., 512, 1, 2 ...). The output of each block is always fed to two directions: Firstly, it is combined with the input of that block and further fed to as input to the next block. Secondly, it is used as a skip-connection; the skip-connections from each block are added together in the end and the result is then fed to the rest of the network.

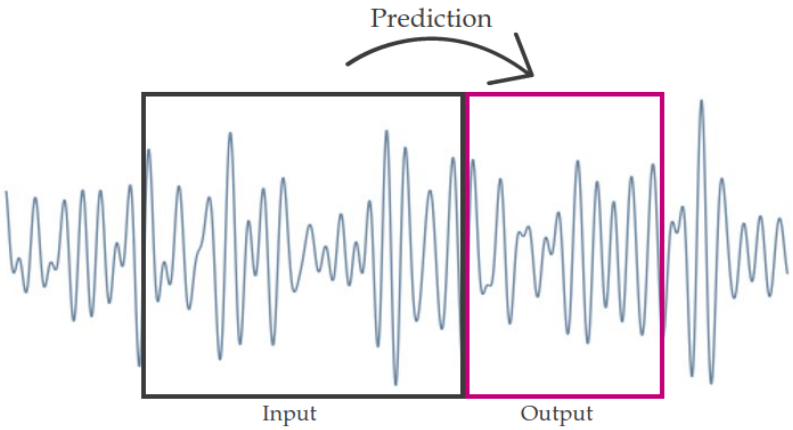
The WaveNet model was able to achieve state-of-the-art results in text-to-speech tasks, significantly improving on the previous methods. Additionally, it was successfully trained to produce music – as a result, it was able to generate believable novel snippets of classical piano music.

2.5 The Present Study

In this thesis we apply deep learning for forward predicting EEG time series. A long-term goal is to eventually employ these techniques in a real time application within a BCI that navigates brain-state dependent TMS.

Based on the extensive previous work done in the field we have a firm ground for presuming that deep learning methods – and especially causal convolutional neural networks – are a promising tool for modeling EEG time series. Importantly, of these approaches the WaveNet model seems the most promising for forward predicting EEG time series as the structure of EEG data is somewhat similar to audio data. Following the previous research, we will in this thesis propose a WaveNet based convolutional neural network model for EEG time series prediction. Furthermore, we will further extend this model into a multivariate version.

How the topic intertwines to the research paradigm of artificial intelligence is threefold. Firstly, it adds knowledge to our attempt at modeling the brain and building devices that simulate intelligence which has been one of the main approaches in the field already since the Dartmouth workshop in 1956. Secondly, machine learning is a standard tool of AI: developing new algorithms and applications will undoubtedly take the field forward. Lastly, the research might also have indirect philosophical consequences as it would add to the ongoing debate in philosophy of AI about free will; is the brain deterministic, if we are able to predict future brain states?



3 Materials and Methods

3.1 Data

The data for this study were retrieved from an open online repository PRED+CT[1] (Cavanagh, Napolitano, Wu, & Mueen, 2017). The chosen dataset ("Depression Rest", accession number d003 (Cavanagh, Bismark, Frank, & Allen, 2019)) is comprised of 121 resting state EEG recordings, each from a different subject. 46 of these subjects are depression or high BDI patients while the rest are healthy participants. For this study we excluded the patient data as well as 3 of the control subjects due to missing or distorted data. Consequently, the remaining 72 recordings from healthy subjects were used to test the models.

The recordings were done with a Neuroscan system with 60 EEG channels and a sampling rate of 500 Hz. Each recording includes both eyes closed and eyes open conditions; 3 minutes for each, totaling up to 6 minutes of data. As eyes closed and open data are known to have different characteristics (see for example Barry, Clarke, Johnstone, Magee, & Rushby, 2007), we chose to only use the eyes open data in this study to introduce consistency to the data.

Apart from band-pass filtering, no preprocessing – such as artifact removal – was done to the data in any of the experiments. The filtering choices for each experiment are explained in section 3.3.

3.2 Models

In this work we developed two deep learning models for forward predicting the EEG time series: a univariate (UVM) and a multivariate (MVM) model. Both of these models predict upcoming values $(t_{n+1}, t_{n+2}, \ldots, t_{n+x})$ of time series of one EEG signal. The UVM does this based on the previous values $(t_1, t_2, \ldots, t_n)$ in the same channel that is to be predicted whereas the MVM additionally processes also the corresponding time series data from other channels.

The UVM is a close adaptation of the audio signal processing WaveNet-model introduced by Oord et al. (2016) (see section 2.4.4). The MVM in turn extends from the UVM by using two dimensional convolutions instead of one dimensional. Both of the models utilise the key concepts of the WaveNet-model: causal convolutions and dilation as well as residual and skip connections. In the following subsections we will describe the architectures of both of the models in detail.

[1] http://predict.cs.unm.edu/

3.2.1 Univariate model

The UVM predicts the upcoming signal in one EEG channel solely based on the preceding signal in the same channel. Figure 5 shows the model architecture.

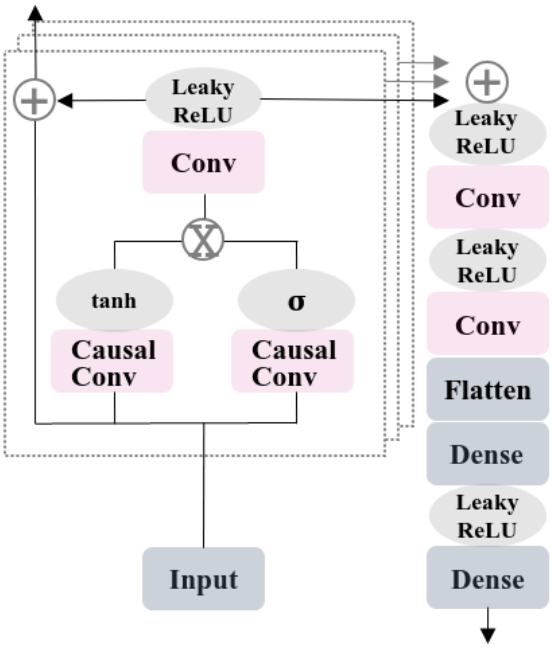


Figure 5: Architecture of the univariate model.

The signal is first led through a stack of blocks containing causal convolution layers. The first block gets as input the original input tensor after which each of the remaining blocks gets as input both the original input tensor as well as the residual signal from the previous block. The outputs of all of the blocks are then added together and the resulting tensor is then passed on to the rest of the network.

There are two main differences between our model architecture compared to the WaveNet model. Firstly, here ReLU activation functions were replaced with LeakyReLU as LeakyReLU seemed to lead to better results compared to ReLU. One reason for this behavior might be the so-called "dying ReLU" phenomenon where a unit with ReLU activation outputs a 0 for every input, thus making that unit inactive, or "dead" (Lu, Shin, Su, & Karniadakis, 2019). This has to do with the property of ReLU where all negative values result in a output of 0 (see section 2.3.1) – so if a value of a unit gets stuck in a negative value, the ReLU activation always turns the output to 0. As a LeakyReLU activation function allows negative values to pass with a small multiplier, it can somewhat escape

the dying ReLU problem.

The second difference is getting the target signal directly as an output instead of a probability distribution specifying the most likely category of the next time stamp. In other words, the softmax layer in the end was removed. Instead, here the last convolutional layer is followed by a flattening layer that does a transformation from a 3-dimensional tensor to a 2-dimensional tensor. After the flattening layer, mapping to the target sequence is done with two fully connected layers.

In the case of EEG data this approach worked significantly better compared to an architecture with softmax. The most likely reason for this is the difficulty of transforming time stamp values of an EEG signal into a small enough amount of categories. In the WaveNet study, possible values of the audio signals are between -1 and 1. For an EEG signal the range on the other hand can be as larg as from -100 to 100. If the EEG signal were to be normalized to the range from -1 to 1, a lot of important information about the amplitude would be lost. If, on the other hand, we used a larger range, a larger amount of categories would also be needed to sufficiently represent that whole range. This, in turn, would significantly increase the computational cost.

3.2.2 Multivariate model

The MVM extends from the UVM by taking additional input from one or more signals in addition to the signal that is being predicted. The additional input signals are all always from the same time range as the original. Figure 6 shows the model architecture.

Inspired by the multivariate CNN design by Liu et al. (2018) we included a univariate processing stage at the very beginning of our model to extract the individual properties of each signal. In addition, there are additional convolution layers for processing the channel dimension inside the blocks.

At the end, after the processing blocks, the target signal is extracted from the tensor and the same output process as in the UVM is performed to said signal.

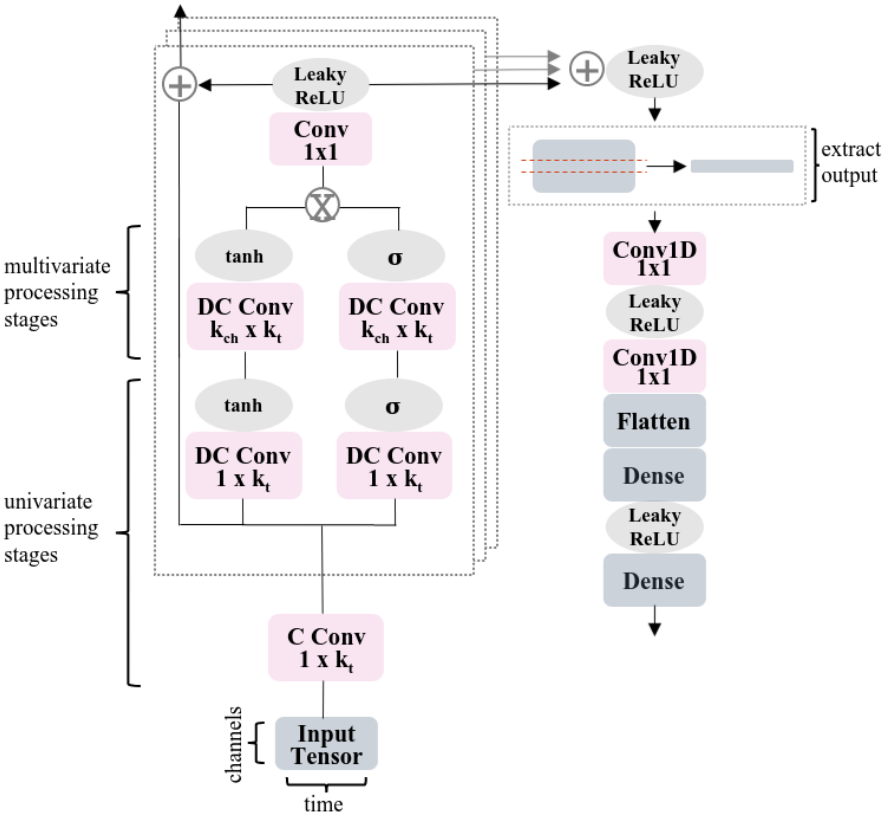


Figure 6: Architecture of the multivariate model. D and C stand for dilated and causal convolution, whereas k_t and k_{ch} refer to the width and height of the convolution kernel.

3.3 Experiments

We performed 3 experiments to evaluate the performances of the univariate and multivariate forward prediction models. Both models were used to predict the C3 channel.

Table 1 shows manually tuned hyperparameters that were applied in each experiment. The parameters were chosen by exploring various options and choosing values that led to best results.

The amount of filters and residual blocks was determined by increasing them until no improvement in mean absolute error occured anymore. The learning rate was adjusted such that it seemed to yield good results in reasonable time.

The Adam optimizer (Kingma & Ba, 2014) was chosen as it has in previous research (see for example Ruder, 2016) proven to be efficient for deep learning problems. Accordingly, He normal was used as kernel initializer because it has shown great performance when used with ReLU activations

Filters	Residual blocks	Optimizer	Learning rate	Kerner initializer
64	12	Adam	0.001	He normal
Loss function	**Epochs**	**Batches**	**Kernel width**	**Kernel height** (multivariate)
Mean absolute error	80	100	9	3

Table 1: Hyperparameters used in training the models.

(He et al., 2015).

The number of epochs was limited to 80 because when more epochs were used, the network started to overfit, i.e. the prediction accuracy on validation set started to decay due to excessive adapting to the training data that deteriorates the generalizability of the model. In each epoch, batch size of 100 was used because batches of 100 do not require too much memory but are still big enough to enable reasonable training.

The kernel width of 9 was used because it enabled modeling sufficiently long sequences: shorter kernels did not perform as well whereas longer kernels significantly increased the computational cost. The height of the kernel for the multivariate model was chosen based on the height of the input: as the input height is 5, a kernel higher than 3 would not be reasonable. A smaller kernel on the other hand would not be sufficient for capturing multivariate information.

3.3.1 Experiment 1: Prediction Performance

In this experiment we examined the prediction performance of both the UVM and MVM. This included analysing the overall prediction accuracy as well as precision of the predicted phase along the whole prediction. In addition, we compared performances of the univariate and multivariate models to see if adding information from neighboring channels improves the accuracy of predictions for C3 channel.

Both model types were trained 50 times. For the training of each model, 60 distinct subjects were chosen at random from the pool of 72 subjects. All data from 50 of these subjects were used as the training data and all data from the remaining 10 subjects were used to test the model. Thus, each of the 100 models that were trained had its own randomly selected training and test sets. In addition, 10 % of the training set was always used for validation during training.

The data were band-pass filtered to α-band (8-12 Hz) before training. The α-band was used in all of the experiments to allow comparability with the study of Zrenner et al. (2018) where the α-band was also used. The filtering was done with a minimum phase finite impulse response filter (FIR) with filter order 825. Order of 825 was used because it is automatically defined based on the size of the transition regions by MNE-Python tool box[2] that was used for preprocessing the data. The minimum phase filter was chosen because of its causal property – with time series prediction it is not possible to use any future time stamps for preprocessing. No other preprocessing was done to the data: as the goal is to use the models in real time analysis, it is crucial to avoid any time consuming extra steps.

For training the UVM, training and testing examples as well as corresponding labels were created by dividing the time series of C3 channel into pieces of 2150 ms with overlap of 1900 ms. The resulting pieces were further divided into input (2000 ms) and target (150 ms) sequences, which correspond to 1000 and 75 timestamps, respectively.

The inputs and labels for training the MVM were created in the same manner, but with one addition: concurrent time series of the four neighboring channels of C3 (C1, C5, FC3, CP3) were added to the input sequences. To be more specific, each single input was a two dimensional matrix (*channels x time*) where the target channel, C3, was placed in the middle (see Figure 7).

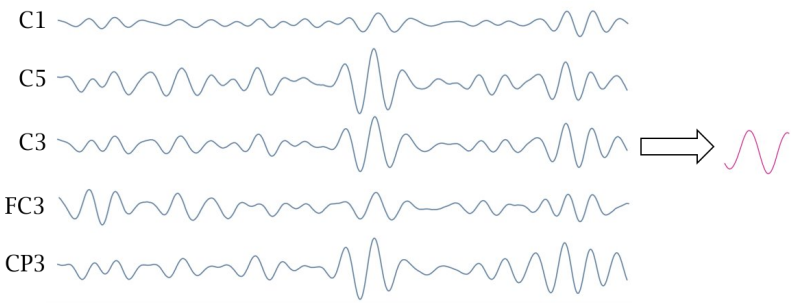


Figure 7: An example of input and label sequences used to train the MVM.

3.3.2 Experiment 2: Comparison to the Autoregressive Model

In experiment 2 we compared our univariate model to the autoregressive (AR) model that is so far the only model that has been used for real time EEG phase estimation (Zrenner et al., 2018). For this comparison we trained a new set of univariate models (UVM2) but with data that what were treated with the same preprocessing procedure as in Zrenner et al. (2018).

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

Thus, we replicated the preprocessing steps introduced in Zrenner et al. (2018). The original process is described in detail in section 2.4.3. In short, Hjorth-C3 signal was formed for sliding window of 500 ms: orthogonal source derivation (Hjorth, 1975) was performed by applying a C3-centered Laplacian operator on C3 channel and its neighboring four channels (C1, C5, FC3, CP3). Next, the signal was band-pass filtered (8-12 Hz, order 128 two-pass FIR filter) and demeaned after which 64 ms from both edges were removed. Finally, AR model coefficients were determined and the signal was predicted 128 ms forward, i.e. from -64 ms to 64 ms.

For this study the training and test data were consequently formed following the aforementioned process with one exception: as our model performs better with a longer input, we used input length of 2000 ms instead of the 436 ms that the AR model uses to determine its parameters. Additionally, also labels for each input sequence were formed as they are required for the training process.

The training and test data time series were divided into sequences of 2256 ms with an overlap of 1900 ms. Then, each sequence was processed following Zrenner et al. (2018): the Hjorth-C3 signal was created and further band-pass filtered and demeaned as well as trimmed 64 ms from both edges. The resulting sequences were finally divided into inputs (2000 ms) and labels (128 ms).

For training and testing the models the data were chosen in the same manner as in the previous experiment: 50 and 10 subjects were picked at random for the training and test data, respectively. Accordingly, 50 models were trained and for each a new set of training and test subjects was selected.

Furthermore, to control for possible differences between our data sets and data of Zrenner et al., we recreated the AR modeling process with our data[3].

3.3.3 Experiment 3: Inter-individual Generalisability

In the final experiment we investigated the inter-individual generalisability of the UVM by examining whether a larger pool of subjects in the train data helps the model to generalise better to unforeseen subjects compared to using fewer subjects in the train data set.

For this, the UVM was trained under 9 conditions. Firstly, the train data sets were composed of data from either 20, 30, 40, 50 or 60 subjects. Secondly, either all of the data from each chosen subject was included or data per subject was limited. In the latter option the amount of data per subject was decreased proportionately as the amount of subjects was increased resulting in evenly sized data sets. In other words, with 20 subjects 100 % of data per subject was used, with 30 subjects 66.7 %, and so on. This was done as we wanted to examine, whether adding more subjects

[3]The code is publically shared in https://github.com/bnplab/phastimate (cited 27.05.2020).

positively impacts generalizability. Machine learning models generally learn the better the more training examples they are provided with, so we would expect that models trained with data of 60 subjects would perform better compared to models trained with 20 subjects simply because of having more training examples and not necessarily because of larger variance within the examples.

Otherwise, this experiment followed the training process of the univariate models in Experiment 1: For each condition the model was trained 50 times, resulting in 450 models in total. Data sets for each model were selected with the same process as in Experiment 1 and each model was again tested on 10 subjects – regardless of how many subjects were included in the train set. Furthermore, band-pass filtering and creating the input and label sequences were perfomed in the same way as in Experiment 1.

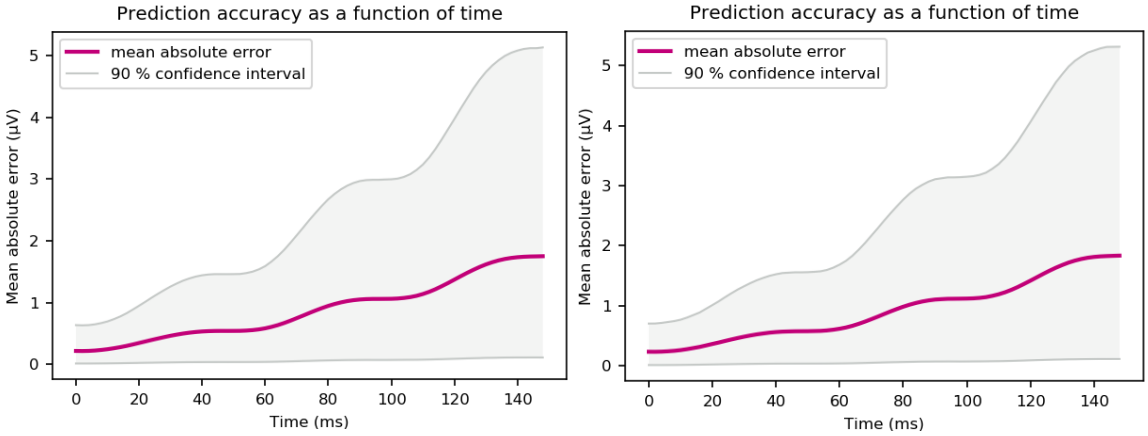
4 Results

4.1 Experiment 1: Prediction Performance

Prediction and phase accuracies for both models were assessed. Prediction accuracy was estimated with mean absolute error of predictions made by the models.

Each model was tested with all of its test data. This was done by moving a sliding window of 2000 ms through each sequence in the test data. At each point the 2000 ms was used as input to the model for predicting the following 150 ms after which the sliding window was always moved 150 ms forward.

For each predicted sequence a mean absoute error for each time stamp was then calculated by comparing the prediction to the true signal. Figure 8 shows mean absolute error and 90 % confidence intervals of the predictions as a function of time for both UVM and MVM.



(a) The prediction accuracy of the 50 UVMs

(b) The prediction accuracy of the 50 MVMs

Figure 8: The progression of mean absolute error along the predicted signal of C3-channel.

We also calculated the means of each prediction's mean absolute error sequence. The means and standard deviations of these means are 0.88 ±0.79 µV (UVM) and 0.92 ±0.79 µV (MVM). The means of the univariate models are thus smaller and this difference is also statistically significant (from a two-sided t-test: T = 28.5, p = 3.6e-179).

For both model types the mean absolute prediction error gradually increases as we go further in the prediction. The mean absolute errors for UVM and MVM are 0.22 and 0.23 in the beginning of the predictions and grow to 1.76 and 1.83 by the end of the predictions. These differences in means

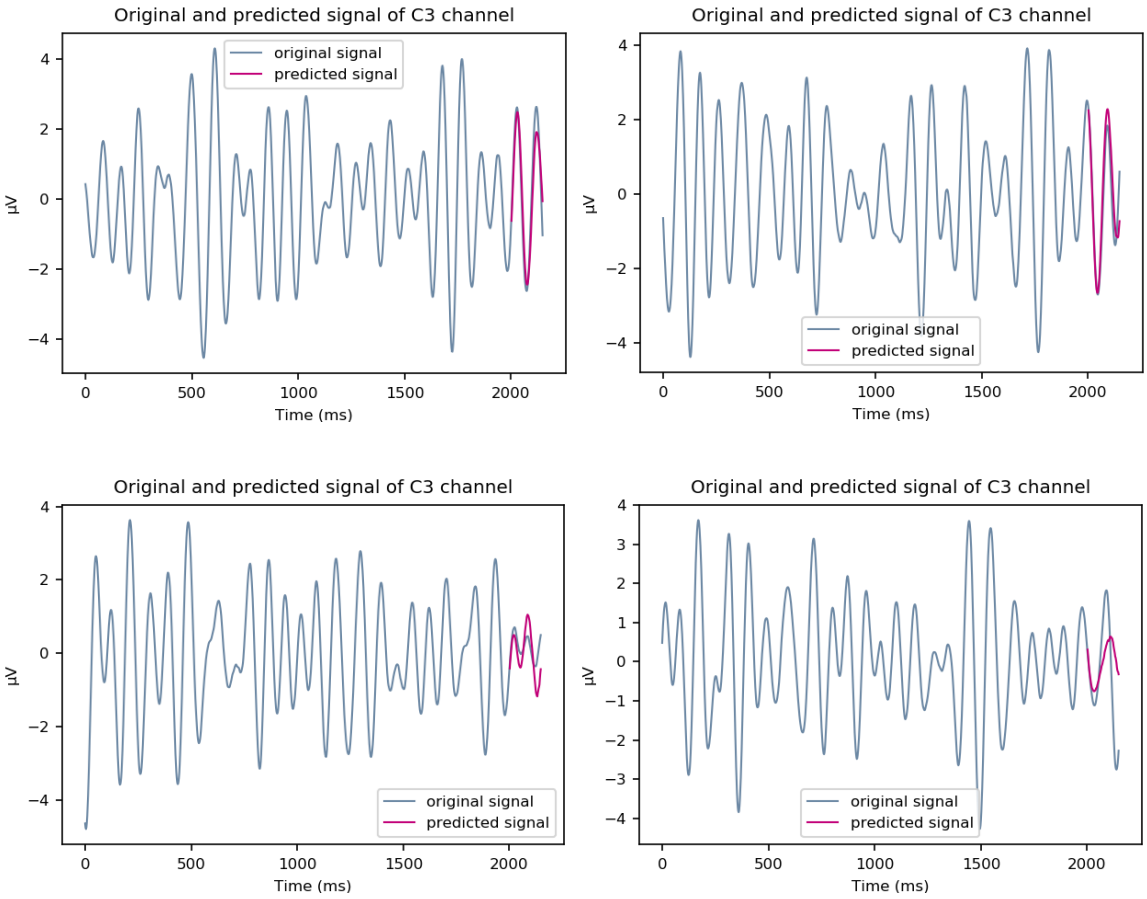


Figure 9: Example predictions made by one of the UVMs.

are statistically significant (from a two-sided t-test: T = 23.3, p = 2.5e-120 (beginning of prediction) and T = 23.3, p = 5.7e-120 (end of prediction)).

In addition to analysing the prediction error as a function of time, we also compared the distribution of the mean absolute errors of the UVM and MVM. Here the mean absolute errors are calculated for the test sets described in section 3.3 such that the error is calculated for the whole sequence rather than separately for each time stamp. One test point represents the mean absolute error of predictions of one model on one of its test subjects, making the total number of test points 500 for both UVM and MVM.

Figure 10 shows the distributions of prediction accuracies of UVM and MVM. The medians of the mean absolute errors are 0.75 µV and 0.80 µV for UVM and MVM, respectively. In addition,

the respective means and standard deviations are 0.88 ±0.46 µV and 0.92 ±0.45 µV. The differences of the means are not statistically significant (from a two-sided t-test: T = 1.3, p = 0.19)

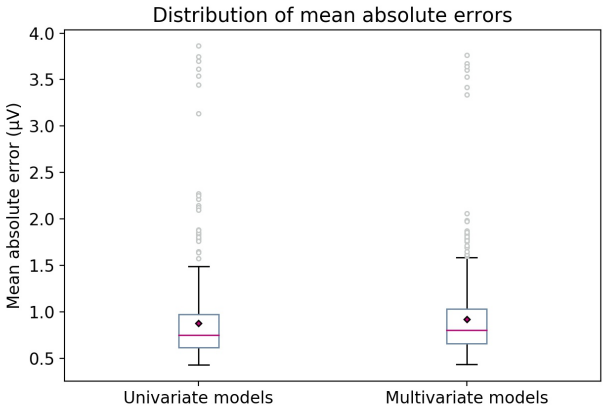
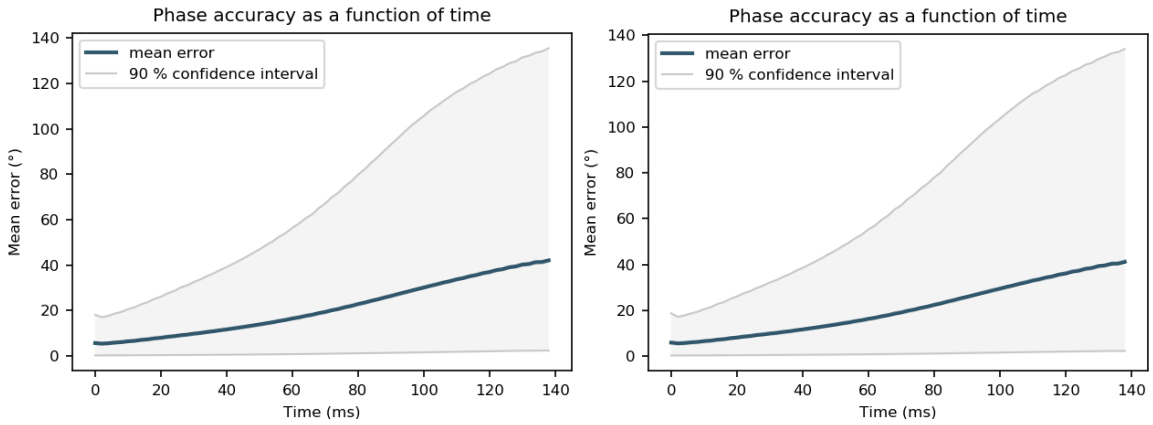


Figure 10: Distributions of mean absolute errors of the predictions of the UVM and MVM. One test point represents mean absolute prediction error of one model on one of its test subjects. The vertical lines denote medians, diamonds means, and circles outliers of the distributions.

The phase errors for both model types were calculated for the same predictions as used in the prediction accuracy estimation. An analytic signal was formed by calculating a Hilbert transform for sequences containing an input and the respective prediction. The input sequence was included in the transform as the Hilbert transform distorts the edges of sequences and we wanted to reliably estimate the phase accuracy from the beginning of the predictions. The phase angle ($\Phi \in [0°, 360°[$) for each time stamp was then calculated from the Hilbert transformed time series after which the inputs were removed from the beginnings of the sequences. The same process was repeated for the original sequences to obtain the true phase at each time point.

Figure 11 shows the mean errors as functions of time for both models. The last 10 ms were removed because of the aforementioned edge distortion. Similarly as the prediction accuracy, also the mean phase accuracy gradually decreases over time, with the mean error starting from 5.68 (UVM) and 5.90 (MVM) and ending at 36.19 (UVM) and 35.56 (MVM). These differences in means are statistically significant (from a two-sided t-test: T = 13.0, p = 8.4e-39 (beginning of prediction) and T = 9.2, p = 4.5e-20 (end of prediction)).

The mean phase errors are 22.44 ±19.18 and 22.04 ±19.02 (UVM and MVM, respectively). Here the mean error of the multivariate models is smaller and this difference is also statistically significant (from a two-sided t-test: T = 12.6, p = 3.0e-36).



(a) UVM (b) MVM

Figure 11: The mean phase errors of the first 140 ms of predicticted signals of the C3-channel.

		mean	median	mean at t = 0	mean at t = 150
Mean absolute error of prediction accuracy (μV)	UVM	0.88 ±0.79	0.69	0.22	1.76
	MVM	0.96 ±0.79	0.72	0.23	1.83
Mean error of phase accuracy	UVM	22.44° ±19.18°	16.48°	5.68°	36.19°
	MVM	22.04° ±19.02°	16.08°	5.90°	35.56°

Table 2: Summary of results of Experiment 1. All of the differences between means of the two models are statistically significant.

4.2 Experiment 2: Comparison to the Autoregressive Model

The prediction performances of the UVM2 and AR models were compared in two ways. Firstly, we assessed the distribution of true phases in those cases when the models predicted that there occured a high or a low peak at time 0. Secondly, the prediction accuracy as a function of time was estimated.

Preprocessing of the data was done as in the training phase. However, the demeaning here was done separately to input and label sequences to avoid using information from future time stamps in input. The input sequences were demeaned using 64 ms of additional data in both edges that

was then trimmed. The label in turn was formed with the following process: 128 ms was added to the end of the same sequence that was used for demeaning the input. That sequence was then demeaned, after which 2064 ms was removed from the beginning and 64 ms from the end. The remaining 128 ms then formed the true label signal.

For the phase estimations, the models were used to make a forward prediction from each time stamp in their test sets. For UVM2 this meant testing each of the 50 models with their respective test sets, each test set containing data from 10 subjects. The AR algorithm in turn was tested with data from 20 subjects. The phase signals from the predictions of the both model types were calculated similarly as in Experiment 1. However, here the Hilbert transform and phase angle calculation were made for solely the predicted sequence, to follow the process in Zrenner et al. (2018).

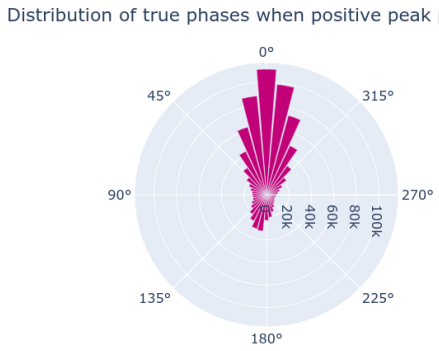
For each prediction it was inspected, whether there occured a positive (0°) or a negative (180°) peak at time 0, i.e. "now"[4]. When a peak was detected, the precise timing of said peak was then determined by performing linear interpolation for the centermost time stamps. The true phase was then estimated from the phase signal of the true signal: linear interpolation was performed to the centermost time stamps to determine the phase at the time of the predicted peak.

Figure 12 shows the distribution of true phases when a high or a low peak is predicted at time 0. Additionally, we calculated the circular means and standard deviations for the true phase distributions. For the high peak condition the circular means of the UVM2 and AR models are 6.08 ±79.15 and 350.20 ±83.64, respectively. The respective values for the low peak condition are 184.94 ±76.82 and 170.33 ±83.65. For both conditions, the difference in means is statistically significant: from Watson-Wheeler test for homogeneity of angles we got W = 2557.1, p = 2.2e-16 (high peak condition) and W = 2997, p = 2.2e-16 (low peak condition).

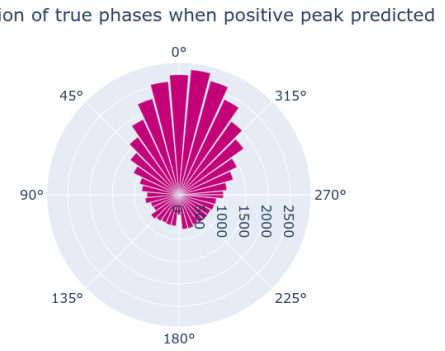
Predictions for determining the mean absolute error as a function of time were done in the same way as in Experiment 1. A sliding window of 2000 ms (500 ms for the AR model) over the test data was used as an input and moved 128 ms forward after each prediction. For the univariate model, all of the test sets of each model were tested and for the AR model, data from all 72 subjects was used.

Figure 13 shows the accuracy of the predictions as a function of time for both of the models. The mean absolute errors here are 0.47 ±0.71 (UVM) and 0.49 ±0.43 (AR). The difference in means here is statistically significant (from a Welch's t-test where population variances are not equal: T = 16.8, p = 3.9e-63).

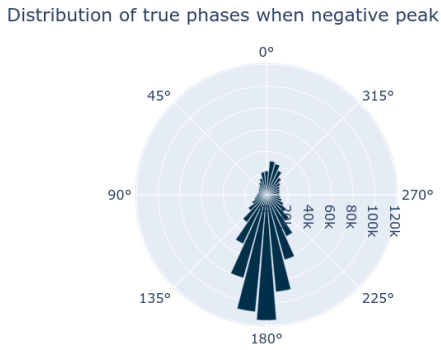
[4]Note that here time 0 is in the middle of the prediction as the prediction starts from -64 ms.



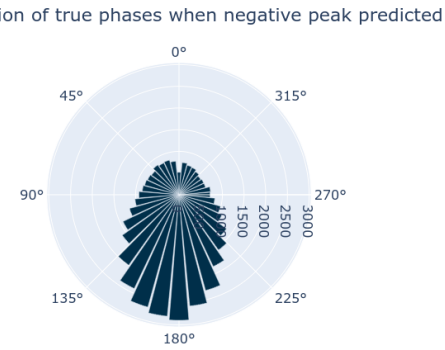
(a) UVM2, high peak condition



(b) AR, high peak condition

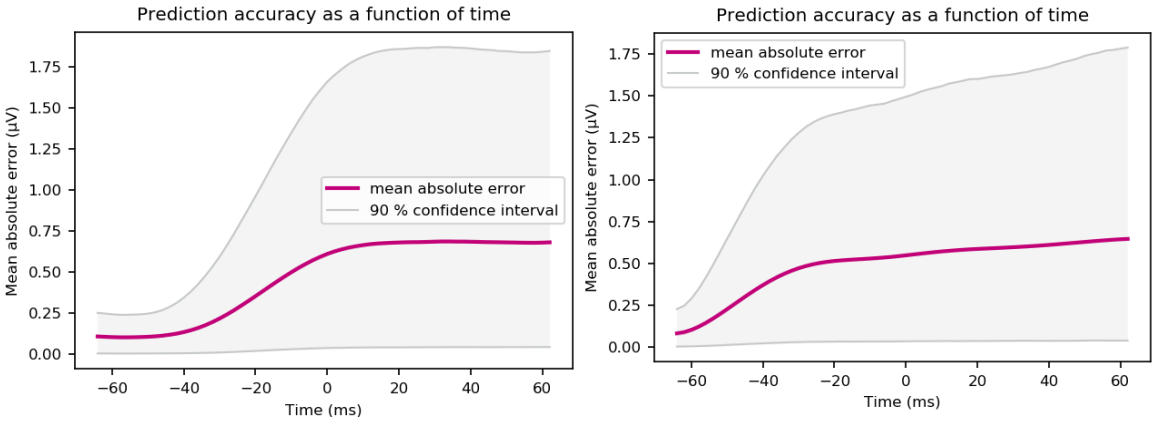


(c) UVM2, low peak condition



(d) AR, low peak condition

Figure 12: Distribution of true phases when an occurence of either a positive or a negative peak at time 0 is predicted by the model. Width of one bar is 10°.



(a) UVM2

(b) AR

Figure 13: The progression of mean absolute error along the predicted signal of channel Hjorth-C3. Although the overall mean absolute error of UVM2 is lower than that of the AR, the mean absolute error at the beginning and end of the prediction is significantly higher for UVM2 (0.11 and 0.68 μV) compared to AR (0.08 and 0.65 μV): from a Welch's t-test where population variances are not equal: T = 34.1, p = 2.6e-255 (beginning of prediction) and T = 14.8, p = 3.0e-49 (end of prediction).

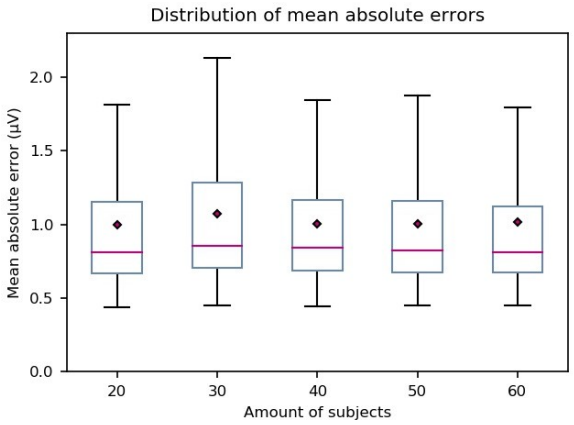
		high peak	low peak
Mean of true phases	UVM2	6.08° ±79.15°	184.94° ±76.82°
	AR	350.20° ±83.64°	170.33° ±83.65°

		mean	median	mean at t = -64	mean at t = 64
Mean absolute error of prediction accuracy (μV)	UVM2	0.47 ±0.71	0.35	0.11	0.68
	AR	0.49 ±0.43	0.41	0.08	0.65

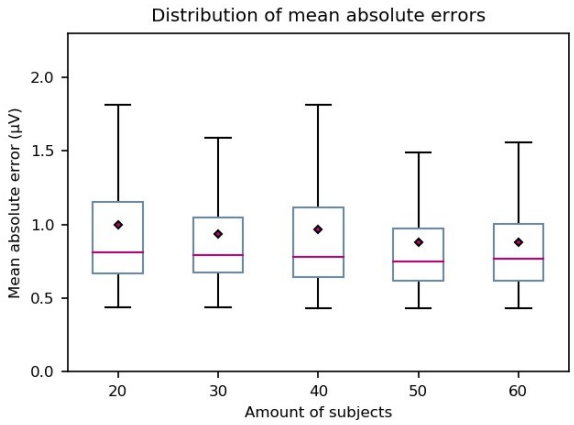
Table 3: Summary of results of Experiment 2. All of the differences in means between the two models are statistically significant.

4.3 Experiment 3: Inter-individual Generalisability

In the generalisability experiment the distributions of the mean absolute errors of two sets of the UVM were tested. Each model (N=450) was tested separately for each of its test subjects as was done in the distribution analysis in Experiment 1. Figures 14a and 14b show the distributions of mean absolute errors of the predictions. The medians, means and standard deviations of the predictions are presented in table 4.



(a) Limited condition



(b) All condition

Figure 14: Distribution of mean absolute errors. Outliers are left out from this figure for improved readability.

Condition		N subjects 20	30	40	50	60
Limited	median	0.81	0.85	0.84	0.82	0.81
	mean	1.00	1.07	1.01	1.01	1.02
	sd	0.54	0.63	0.53	0.51	0.61
All	median	0.81	0.79	0.78	0.75	0.77
	mean	1.00	0.93	0.97	0.88	0.88
	sd	0.54	0.45	0.60	0.46	0.40

Table 4: Medians, means and standard deviations of the models. Unit of medians and means is µV. The training data sets in Limited condition are even sized whereas the size of the data sets in All condition increase as subjects are added. The means in the All condition are significantly different from each other (from one-way ANOVA test: $F = 5.6$, $p = 0.00017$) whereas the differences in the Limited condition are not (from one-way ANOVA test: $F = 1.4$, $p = 0.22$).

5 Discussion

Current practice in brain-state dependent stimulation is to use an autoregressive forward prediction model for anticipating upcoming electrical activity of the brain (Zrenner et al., 2018). In this work, we explored the possibility of improving on this by using deep learning. For this purpose, we introduced two convolutional neural network models for forward predicting EEG time series. Performances of these models were evaluated and compared to the autoregressive model.

In the first experiment we examined the prediction and phase estimation accuracies of the UVM and MVM. The results suggest that deep learning indeed is a feasible approach for modeling how the EEG time series develops.

Contrary to our original hypothesis, adding data from neighboring channels did not improve the prediction accuracy. There might be several reasons for this. Firstly, signals from adjacent EEG channels are extremely similar to one another. Hence, it might be the case that they do not provide sufficient additional information on the development of their neighboring channels. Later, the MVM could thus be improved by using a more suitable set of input channels. Secondly, the MVM architecture proposed here might fail to extract some or all of the adequate multivariate features needed to provide sufficient additional information about the progression of the centering channel. In further work, different model architectures or hyperparameters could be tested to improve MVM's prediction accuracy.

However, despite failing to improve on the UVM's prediction accuracy, the MVM did show slightly better performance in predicting the phase of the signal compared to the UVM. Thus, it seems that the MVM succeeded in capturing the frequency of the signal slightly better than the UVM. This suggests that the architecture of the MVM did nonetheless manage to extract some useful multivariate information. Hence, a different set of input channels rather than a different model architecture might be a better way to improve the prediction accuracy. A more informed selecion of input channels could be made for example by localizing the sources of the signal we are predicting – by performing source estimation, we can combine EEG time series information with functional and structural connectivity information from magnetic resonance imaging.

The deep learning models were always tested on data from subjects that were not included in training the model although an individually trained model would most likely lead to best results. However, generalizability across subjects is a beneficial feature for the closed-loop TMS application. This is because of two reasons: the amount of data needed for training the model and the time required for the training process. In this study, EEG recordings worth of 150 minutes were used to

train each model and each training took approximately one hour – recording sufficient amount of data and training the model every time the device is used with a new subject would probably not be feasible as the process would last for over three hours.

In the second experiment the UVM2 was compared to the AR model. Here, we obtained a clear improvement on the phase estimation task – the true phases in both high and low peak conditions are less precisely estimated when the peaks are predicted with the AR model compared to our UVM2. As such, we can say that deep learning can be used to improve on the existing forecasting practice.

It is to be noted that the performance of the AR model here is a bit worse compared to its performance in previous research (Zrenner et al., 2018). The most likely reason for this is that in Zrenner et al. (2018), only predictions spanning a given power threshold were included in the analysis whereas here all of the predictions were included.

The mean prediction accuracy of the UVM2 was only slightly better compared to that of the AR model and at some points, for instance, in the beginning and end of the predictions, the mean absolute error of the AR model was even fractionally smaller. However, the UVM2 shows better performance in the beginning of the prediction as the prediction accuracy of the AR model decreases more quickly compared to the UVM2.

Overall, we can conclude that the deep learning approach is especially good at capturing the signal frequency whereas there is still room for improving on modeling the signal amplitude. In future research, the multivariate approach with a more carefully selected set of input channels might be useful here as it can take into consideration how the signals propagate in the brain.

Finally, in the third experiment we examined whether the generalisability ability of the UVM is dependent on how many subjects the training data is constituted of. Increasing the pool of subjects did not show any effect on the generalizability here. This suggests that only the amount of data matters and not from whom it is recorded.

Although the results presented in this study show great promise for deep learning being a good approach for predicting EEG signals, there are still a few limitations that should be addressed in future research before these models can be implemented as a fully functioning part of a closed-loop brain stimulation device. Firstly, the models here were used to predict solely data from healthy subjects with eyes open data. For clinical use it might be necessary to do adjustments to achieve the same performance on patient data as well as with eyes closed data. Secondly, only the signal of C3-channel in one frequency band was predicted here – for a multi-locus TMS system we need to predict all of the signals in various frequency bands. Lastly, empirical testing is needed to verify the

suitability of the deep learning models for a real time prediction application.

As a conclusion we can say that deep learning and in particular convolutional neural networks are a good approach for the problem at hand. All in all, it was shown here that, especially in phase estimation, our models are able to produce state-of-the-art results in forward predicting EEG time series.

References

Babu, G. S., Zhao, P., & Li, X.-L. (2016). Deep convolutional neural network based regression approach for estimation of remaining useful life. In *International conference on database systems for advanced applications* (pp. 214–228). Springer.

Bai, S., Kolter, J. Z., & Koltun, V. (2018). Convolutional sequence modeling revisited. Retrieved from https://openreview.net/forum?id=rk8wKk-R-

Barry, R. J., Clarke, A. R., Johnstone, S. J., Magee, C. A., & Rushby, J. A. (2007). EEG differences between eyes-closed and eyes-open resting conditions. *Clinical neurophysiology*, *118*(12), 2765–2773.

Biasiucci, A., Franceschiello, B., & Murray, M. M. (2019). Electroencephalography. *Current Biology*, *29*(3), R80–R85.

Blinowska, K. J. & Malinowski, M. (1991). Non-linear and linear forecasting of the eeg time series. *Biological cybernetics*, *66*(2), 159–165.

Borovykh, A., Bohte, S., & Oosterlee, C. W. (2017). Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691*.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2008). *Time series analysis: Forecasting and control* (4th ed.). Hoboken, New Jersey: John Wiley & Sons.

Buch, E., Weber, C., Cohen, L. G., Braun, C., Dimyan, M. A., Ard, T., ... Fourkas, A., et al. (2008). Think to move: A neuromagnetic brain-computer interface (BCI) system for chronic stroke. *Stroke*, *39*(3), 910–917.

Cavanagh, J. F., Bismark, A. W., Frank, M. J., & Allen, J. J. (2019). Multiple dissociations between comorbid depression and anxiety on reward and punishment processing: Evidence from computationally informed eeg. *Computational Psychiatry*, *3*, 1–17.

Cavanagh, J. F., Napolitano, A., Wu, C., & Mueen, A. (2017). The patient repository for EEG data + computational tools (PRED+CT). *Frontiers in neuroinformatics*, *11*, 67.

Craik, A., He, Y., & Contreras-Vidal, J. L. (2019). Deep learning for electroencephalogram (EEG) classification tasks: A review. *Journal of neural engineering*, *16*(3), 031001.

Cui, Z., Chen, W., & Chen, Y. (2016). Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995*.

Donati, A. R., Shokur, S., Morya, E., Campos, D. S., Moioli, R. C., Gitti, C. M., ... Pereira, G. A., et al. (2016). Long-term training with a brain-machine interface-based gait protocol induces partial neurological recovery in paraplegic patients. *Scientific reports*, *6*, 30383.

Doucoure, B., Agbossou, K., & Cardenas, A. (2016). Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data. *Renewable Energy*, *92*, 202–211.

Gharabaghi, A., Kraus, D., Leao, M. T., Spüler, M., Walter, A., Bogdan, M., ... Ziemann, U. (2014). Coupling brain-machine interfaces with cortical stimulation for brain-state dependent stimulation: Enhancing motor cortex excitability for neurorehabilitation. *Frontiers in human neuroscience*, *8*, 122.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge, Massachusetts: MIT press.

Hallett, M. (2000). Transcranial magnetic stimulation and the human brain. *Nature*, *406*(6792), 147.

Haykin, S. (2010). *Neural networks and learning machines* (3rd ed.). Upper Saddle River, New Jersey: Pearson Education.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the ieee international conference on computer vision* (pp. 1026–1034).

Hernández, J. L., Valdés, J., Biscay, R., Jiménez, J. C., & Valdés, P. (1995). EEG predictability: Adequacy of non-linear forecasting methods. *International journal of bio-medical computing*, *38*(3), 197–206.

Hjorth, B. (1975). An on-line transformation of EEG scalp potentials into orthogonal source derivations. *Electroencephalography and clinical neurophysiology*, *39*(5), 526–530.

Hotson, G., McMullen, D. P., Fifer, M. S., Johannes, M. S., Katyal, K. D., Para, M. P., ... Wester, B. A., et al. (2016). Individual finger control of a modular prosthetic limb using high-density electrocorticography in a human subject. *Journal of neural engineering*, *13*(2), 026017.

Ibagon, G., Kothe, C., Bidgely-Shamlo, N., & Mullen, T. (2018). Deep neural networks for forecasting single-trial event-related neural activity. In *2018 ieee international conference on systems, man, and cybernetics (smc)* (pp. 1070–1075). IEEE.

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., & Wu, Y. (2016). Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kose, U. & Arslan, A. (2017). Forecasting chaotic time series via anfis supported by vortex optimization algorithm: Applications on electroencephalogram time series. *Arabian Journal for Science and Engineering*, *42*(8), 3103–3114.

Kraus, D., Naros, G., Bauer, R., Khademi, F., Leão, M. T., Ziemann, U., & Gharabaghi, A. (2016). Brain state-dependent transcranial magnetic closed-loop stimulation controlled by sensorimotor desynchronization induces robust increase of corticospinal excitability. *Brain stimulation*, *9*(3), 415–424.

Krucoff, M. O., Rahimpour, S., Slutzky, M. W., Edgerton, V. R., & Turner, D. A. (2016). Enhancing nervous system recovery through neurobiologics, neural interface training, and neurorehabilitation. *Frontiers in neuroscience*, *10*, 584.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lefaucheur, J.-P., Aleman, A., Baeken, C., Benninger, D. H., Brunelin, J., Di Lazzaro, V., ... Hummel, F. C., et al. (2020). Evidence-based guidelines on the therapeutic use of repetitive transcranial magnetic stimulation (rTMS): An update (2014–2018). *Clinical neurophysiology*, *131*(2), 474–528.

Lefaucheur, J.-P., André-Obadia, N., Antal, A., Ayache, S. S., Baeken, C., Benninger, D. H., ... De Ridder, D., et al. (2014). Evidence-based guidelines on the therapeutic use of repetitive transcranial magnetic stimulation (rTMS). *Clinical Neurophysiology*, *125*(11), 2150–2206.

Liu, C.-L., Hsaio, W.-H., & Tu, Y.-C. (2018). Time series classification with multivariate convolutional neural network. *IEEE Transactions on Industrial Electronics*, *66*(6), 4788–4797.

Lu, L., Shin, Y., Su, Y., & Karniadakis, G. E. (2019). Dying ReLU and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733*.

Mitchell, T. (1997). *Machine learning*. New York: McGraw-Hill Higher Education.

Mohammed, M., Khan, M. B., & Bashier, E. B. M. (2016). *Machine learning: Algorithms and applications*. CRC Press.

Moritz, C. T., Perlmutter, S. I., & Fetz, E. E. (2008). Direct control of paralysed muscles by cortical neurons. *Nature*, *456*(7222), 639–642.

Pulido, M., Melin, P., & Castillo, O. (2014). Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the mexican stock exchange. *Information Sciences*, *280*, 188–204.

Ren, C.-x., Wang, C.-b., Yin, C.-c., Chen, M., & Shan, X. (2013). The prediction of short-term traffic flow based on the niche genetic algorithm and bp neural network. In *Proceedings of the 2012*

international conference on information technology and software engineering (pp. 775–781). Springer.

Roy, Y., Banville, H., Albuquerque, I., Gramfort, A., Falk, T. H., & Faubert, J. (2019). Deep learning-based electroencephalography analysis: A systematic review. *Journal of neural engineering*.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Samanta, B. (2011). Prediction of chaotic time series using computational intelligence. *Expert Systems with Applications*, *38*(9), 11406–11411.

Stojov, V., Koteli, N., Lameski, P., & Zdravevski, E. (2018). Application of machine learning and time-series analysis for air pollution prediction.

Sugihara, G. & May, R. M. (1990). Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature*, *344*(6268), 734–741.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... Kavukcuoglu, K. (2016). WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

van den Oord, A., Kalchbrenner, N., & Kavukcuoglu, K. (2016). Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*.

Yang, J., Nguyen, M. N., San, P. P., Li, X. L., & Krishnaswamy, S. (2015). Deep convolutional neural networks on multichannel time series for human activity recognition. In *Twenty-fourth international joint conference on artificial intelligence*.

Zaytar, M. A. & El Amrani, C. (2016). Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. *International Journal of Computer Applications*, *143*(11), 7–11.

Zhou, T., Gao, S., Wang, J., Chu, C., Todo, Y., & Tang, Z. (2016). Financial time series prediction using a dendritic neuron model. *Knowledge-Based Systems*, *105*, 214–224.

Zrenner, C., Desideri, D., Belardinelli, P., & Ziemann, U. (2018). Real-time EEG-defined excitability states determine efficacy of TMS-induced plasticity in human motor cortex. *Brain stimulation*, *11*(2), 374–389.