Master's Thesis

Radboud University

Computational Cognitive Neuroscience Lab
Faculty of Social Science

Automatic Sleep Stage Classification using Convolutional Neural Networks with Long Short-Term Memory

by

Simon Johannes Kern

in partial fulfilment of the

requirements for the degree of

Master's of Artificial Intelligence

Supervisors:

Frederik Weber, University of Tübingen

Umut Güçlü, Radboud University

Dr. Marcel van Gerven, Radboud University

Radboud University, Nijmegen

August 18, 2017

Acknowledgement

I would like to thank Frederik for his exellent supversion as well as Marcel and Umut for their useful input and tips towards the technical implementation of the thesis. Furthermore I want to thank my friends and office mates who were doing a good job of keeping me sane during the last nine months. Last but not least I want to thank my father and sister who were always there for me and gave me lots of support throughout my academic endeavours.

Abstract

The division of sleep into different stages using EEG signals is a commonplace practice in sleep laboratories and an indispensable tool for clinicians and researchers. Despite the advances in artificial intelligence, the sleep stage scoring process is in most cases still performed manually. As the scoring process is tedious and time-consuming, its automatization is desirable. In this study a convolutional neural network is trained to automatically extract features from raw 30-second epochs of the EEG, EMG and EOG. An extension of the network using long short-term memory is used to make a sleep stage prediction given the six preceding epochs. To validate the automatic feature extraction a comparison with a hand-crafted feature extraction approach using 37 features is made. The networks were trained using the first 50 records of the public CCSHS dataset and further validated on the public Sleep-EDFx dataset, the public UCD dataset, the private EMSA dataset, as well as records 50 to 100 of the CCSHS. Results show that the network is able to achieve state-of-the-art performance on the CCSHS (record 0-50, accuracy 89%, F_1 81%). Furthermore, without retraining, the network successfully recognizes sleep stages on unseen data of a similar cohort (CCSHS record 50-100: accuracy 91%, F_1 84%, Sleep-EDFx accuracy 81%, F_1 72%, EMSA 83% F_1 72%). Performance were consistently higher compared to the hand-crafted feature approach. The results demonstrate that automatic feature extraction on sleep data is possible and learns features similar to the ones described in the sleep scoring manual of the American Academy of Sleep Medicine.

Contents

List of Figures

All figures are self-created if not indicated otherwise.

List of Tables

1 Introduction

1.1 Overview

The division of sleep into different stages using electroencephalogram (EEG) signals is a commonplace practice in sleep laboratories that has changed little since its formal introduction in the 1960s (Rechtschaffen and Kales, 1968). This division is indispensable for clinicians, e.g. for the diagnosis of sleep related disorders, as well as for sleep researchers aiding the discovery of the mechanisms and functions of sleep(Barkoukis and Avidan, 2007). The sleep stage scoring (also: sleep staging) process is most of the time still performed manually by a trained technician and therefore very tedious and time-consuming (Iber et al., 2007) Although, many papers have been published which propose automatic sleep stage detection systems none of them could assert themselves as an industry standard (Hamida and Ahmed, 2013). Main reasons for that are their unreliability as well as the high costs of the software (Penzel et al., 2003). Up to this day, no open-source implementation or package is available for the researchers or clinicians. In this paper, I will give an overview to the state-of-the-art of automatic sleep scoring and propose an artificial neural network architecture for the detection of sleep stages. I will compare a novel automatic-feature-extraction approach using a convolutional neural network to an approach using hand-crafted features and a feed forward neural network. Additionally, I will combine each approach with an LSTM to test whether modelling the temporal nature of the sleep stages improves performance. This thesis works as a basis towards producing an open-source Python package with an easy-to-use interface aimed at clinicians and researchers.

1.2 Sleep and how it is scored

Sleep is defined as a period of decreased responsiveness that all higher organisms undergo within their daily routine (Cirelli and Tononi, 2008). While the purpose of sleep is not yet fully understood, strong evidence can be found for the importance of sleep for memory consolidation and neuronal regeneration (Lee-Chiong, 2005; Rasch and Born, 2013). Sleep quality can be linked to many health-related issues such as cardiovascular diseases, diabetes or obesity (Jackson et al., 2013; Miller and Cappuccio, 2007). Sleep disorders also have the highest comorbidity with almost all psychiatric disorders (Franzen and Buysse, 2017). Among all mammals, sleep periods vary from 2 to 20 hours with humans needing 7.5 hours of sleep on average per night (Campbell and Tobler, 1984; Carskadon et al., 2005). During sleep different phases of brain activation can be distinguished. The current standard published by the American Academy of Sleep Medicine (AASM) (Iber et al., 2007) differentiates between four different sleep stages: S1 (drowsiness), S2 (light sleep), S3/SWS (Slow-Wave Sleep) and REM (Rapid Eye Movement sleep). The AASM standard is a revision of the older standard rules by Rechtschaffen and Kales (1968). The older standard contained sleep stage 4 which was combined with stage 3 in the update of the recommendation due to their high similarity. One sleep cycle lasts for around 90 minutes and comprises the phases stage 2 (S2), slow-wave sleep (SWS) and rapid eye-movement (REM) sleep while stage 1 (S1) is normally found while falling asleep or after arousals during the night. Typically 3-5 such cycles are completed throughout the night. Figure 1 shows an idealized scoring throughout a night, a so-called hypnogram.

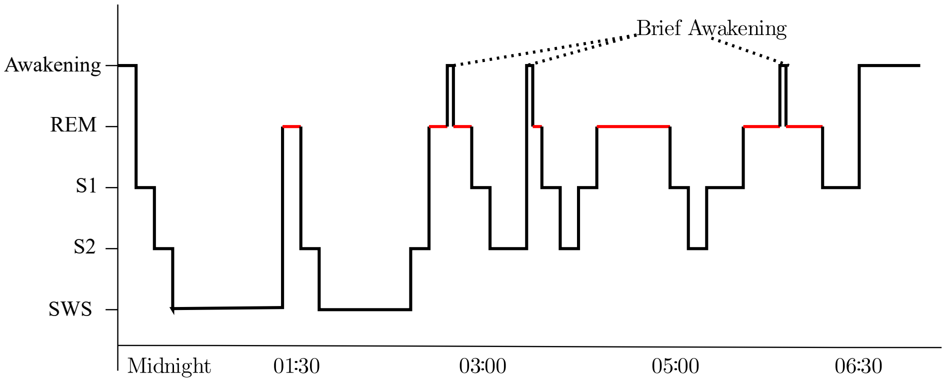


Figure 1: Idealized hypnogram of a typical night with 6.5 hours of sleep[1]

These sleep stages are defined by associated patterns in the EEG, electromyogram (EMG), and electrooculogram (EOG). The common practice for scoring sleep is that a sleep physician (a somnologist) or sleep lab nurse manually goes through the entire night and rates the sleep stage for each 30 second epoch of the polysomnographic signals according to the visual classification rules. This tedious and time-consuming process takes an experienced sleep scorer around 30-90 minutes for an 8-hour recording. As most sleep laboratories have multiple patients per night, the evaluation of recordings can therefore be considered an expensive economic factor. Furthermore, the inter-rater agreement rate of human scorers is only around 80% (Danker-Hopfe et al., 2009) providing a significant source of error for the evaluation of sleep recordings and among scientific studies.

[1] Adapted from `https://commons.wikimedia.org/wiki/File:Sleep_Hypnogram.svg`

1.3 Automatization of Sleep Scoring and Problems of the Scoring Standard

Many approaches have been proposed that try to automatize the sleep scoring procedure. But these approaches suffer from the following problems and shortcomings:

- Lack of large training databases: Sleep data acquisition is costly and time intensive. The quality of the automatization depends largely on the amount of available data for training a classifier. Only few public datasets exist and often researchers have used the same small datasets such as the Sleep-EDFx from PhysioNet (Goldberger et al., 2000) for creating or training their classifier models. Larger datasets have only been made available in the recent years (Dean et al., 2016).

- Inter-subject variability: Sleep EEG of the same sleep stage can look very different depending on the age, present sleep disorders and other neurological diseases. As most datasets contain mostly healthy subjects of a certain age group only, the classifiers are often not able to transfer their performance to other groups and lack of public datasets for such groups prevent further investigation of the problem.

- Inter-rater variability: Large variance in scoring style persists, leading to a lack of consistent ground truth labels (Silber et al., 2007). Only few datasets have a consensus scoring of multiple scorers. Even the intra-rater reliability is only around 0.79 (Cohens kappa) (Whitney et al., 1998; Silber et al., 2007).

Additionally, there are limitations of the current manual scoring standards that transfer to the quality of automatic sleep scoring:

- Arbitrary duration of epochs: The scoring in 30 second epochs has no biological basis. The main reason for this is that when EEG was still recorded analogue on paper, exactly 30 seconds of EEG would fit on a page.

- Validity of the sleep scoring: It is not clear to what extent the existing categorization correlates to most biological processes. The phases were mainly defined empirically and while some sleep phases (SWS, REM) possess discernible characteristics, the distinction between S1 and stages such as Wake and Stage 2 are less clear. Therefore, most scoring disagreement is found between these sleep stages (Rosenberg and Van Hout, 2014; Silber et al., 2007) (23-74% agreement for S1). Furthermore, it might be possible that even more phases can be clearly and conceptually distinguished (e.g. tonic vs. phasic REM

(Ermis et al., 2010; Simor et al., 2016)) or another separation is more suitable (Lo and Hi Deep Sleep (Onton et al., 2016) as separators of non-REM sleep).

- The current sleep state of each brain region can differ from its neighbouring regions: Recent research found out that sleep is not only a whole-brain phenomenon (Nir et al., 2011). Despite being closely correlated, different brain regions can transition between sleep stages with a delay of sometimes several minutes (Magnin et al., 2010). A probabilistic scoring could therefore be appropriate to capture these phase delays as well as the uncertainty about the whole brain state.

Nevertheless, the automatization of the scoring process would have clear benefits:

- The scoring would be consistent. An algorithm will produce the same results each time it runs, which is not the case for humans (Whitney et al., 1998)
- Researchers and clinicians would save a significant amount of time that could be reallocated to more useful tasks

1.4 An Overview of Automatic Sleep Stage Classification

Approaches to automatize the sleep scoring process can be traced back to the early age of computer based analysis in the late 1960s (Itil et al., 1969). Despite advances in machine learning over the past decades, no automatic sleep stage algorithm has been able to establish itself as an industry standard neither in research nor in a medical application. Reasons for this include the lack of trust in algorithms by clinicians, poor reliability of some systems and bad performance on pathological sleep (see also section 1.3). Up to this point many methods have been proposed to automate the sleep scoring process. These can be divided into three main classes:

(I) Hand-crafted features and expert-knowledge-driven set of rules

(II) Hybrid approaches with hand-crafted features and machine learning classification

(III) Machine intelligence for automatic feature extraction and classification.

The first part of the paper will focus on reviewing recent publications about automatic sleep stage classification starting at 2013 as well as mentioning older types of approaches. As this thesis focuses on creating a class (III) system, extra attention will be given to these types of systems. A review for publications before 2013 can be found in Hamida and Ahmed (2013).

1.4.1 Class (I) - Rule Based Expert Systems

Algorithms of this class consist of rules made by experts which are executed by the machine. Examples could be: “If amplitude>40, stage = SWS”. While class (I) systems can only be found in the earlier approaches to automated analysis, their advantage is that the classification stays transparent in each aspect of the process. Every decision can be traced back to an understandable rule. This enables full control over the system. A disadvantage is that the chosen rules might be suboptimal and are usually very simple. The manual construction of rules is not applied anymore in modern research and is merely mentioned for historical reasons. Recently only (Liang et al., 2016) can be regarded a form of this kind of system. It uses a genetic algorithm to find an optimal set of nine rules that classifies sleep stages using eight spectral and temporal features (accuracy: 88%, kappa: 0.82).

1.4.2 Class (II) - Machine Learning on Hand-crafted Features

In this approach, features defined by experts are fed into a machine learning algorithm like a decision tree or a k-nearest-neighbours (KNN) classifier. Typical features consist of the spectral bands of brain activity or mean power of the signal. The classifier tries to find an optimal mapping from the input features to the class labels. Some classifiers are able to give an importance ranking for the input features while others are not. One advantage is the adaptiveness of the classifier which outperforms rule-based methods in most cases and the simplicity of the classification. As many classifiers perform similarly well, their performance relies heavily on the quality of the features and the pre-processing of the data. Often, additional abilities are added to the algorithms, such as an artefact rejection or a form of temporal smoothing for a succession of sleep stages.

Most features defined by experts can be categorized into three categories:

Spectral features: Frequency components are extracted from the signal using methods such as Fourier or wavelet transformation. Information processing in the brain seems to be correlated to different oscillatory processes. Brain states such as sleep stages are characterized by distinct frequencies, such as slow waves in the delta band (0.5-4 Hz) that give slow-wave sleep its name. Most often used are frequencies from 0.5-50 Hz which are be divided into delta (0.5-4 Hz), theta (4-8 Hz), alpha (8-12 Hz), beta (12-40 Hz) and gamma (40-100 Hz) and sometimes an additional band for sleep spindles (12-16 Hz).

Temporal features: Temporal features try to account for the change of brain activation over time. As each sleep stage depends on the previous brain states, the inclusion of temporal features provides important information about the classification. Examples are change in peak amplitude as an indicator of SWS or relative change of frequencies.

Statistical features: The signal can often be described by simple statistical properties such as the signal minimum and maximum or the number of zero crossing, for instance to detect eye movements. Other statistical measures are median signal value, standard deviation of the kurtosis and skewness of the signal.

Complexity features: The complexity of the signal is a good measure of the general level of activity in the brain. From an information theoretical perspective it measures, how many bits are minimally necessary to express the same content in a compressed form. Slow-wave sleep has far less complex signals than for instance wakefulness.

Class (II) systems still make up for the majority of published papers. Besides the mentioned features, a multitude of other forms of features can be included such as features based on geometric measures as well as a combination of spectral and temporal features with e.g. wavelet analysis. Although every paper published a new combination of classifiers and features, the results are often very similar which seems to suggest that no optimal set of features and classifiers exist. One example of such a system given by Tsinalis et al. (2016a) who uses Morlet wavelets to extract time-frequency features which are then used to train a stacked sparse auto-encoder, a type of ANN. Because of the strong class imbalance between the sleep stages, they train the classifier with class-balanced randomly sampled batches. This allows for a much higher accuracy in the smaller classes such as S1 (overall accuracy 78%, mean F_1 score 84%). A nice comparison between classifiers and features is given by Sen et al. (2014). They extract a wide selection of 41 features including spectral, temporal, entropy, statistical, linear and non-linear as well as other types. A feature selection analysis is performed using different methods and five different classifiers: random forests, neural networks, decision trees, support vector machines and radial basis function networks. They conclude that the random forest had the best performance with the least amount of necessary features and report an overall accuracy of a one-versus-all classification of 97%. This one-versus-all classification is, however, not a very good measure of performance as it does not say anything about the desired multi-class confusion for creating a hypnogram.

1.4.3 Class (III) Automatic Feature Extraction

The most recent trend in machine learning is the automatic retrieval of features. An algorithm (most often an ANN) is given a dataset in which it should find distinctive patterns by itself. These techniques are sometimes also able to learn features in an unsupervised manner, without pre-labelled data (e.g. using a Deep Belief Network, DBN). This has the advantage that an optimal set of features can be found by the network without a human bias. A disadvantage is that these features can be abstract and incomprehensible for humans and obscure the decision process (black-box process). These methods are either combined with another classifier or are able to perform a classification intrinsically. While the vast majority of literature reports class (II) systems, in the recent years automatic feature extraction (III) has gained popularity. A more detailed description and a general introduction to ANNs can be found in section 1.5.

As sleep stages follow certain transition patterns, most approaches try to account for these temporal dependencies by using temporal models. In their simple version, a hidden-Markov-model (HMM) can be seen as a temporal smoothing operator, biasing the stage decision based upon the previous epoch. Various extensions for longer dependencies than one epoch are available such as conditional neural fields (CNF) or long-short-term memory (LSTM) neurons.

Up to this point, five papers discuss the automatic extraction of features of sleep EEG using ANNs. Längkvist et al. (2012) use a DBN which tries to hierarchically model the data distribution using stacked restricted Boltzmann machines with moderate success (accuracy: 67%, F_1: 36-83%). The method of Zhang et al. (2016) can be seen as an extension of that using a variant called Sparse Deep Belief Networks (SDBN) which enforces a sparse activation of neurons followed by an ensemble of Support-Vector-Machines (SVM) and a KNN classifier with good results. Furthermore, their algorithm includes a HMM to account for the temporal dependency and achieves human-level recognition (accuracy 91%, F_1: 86-96%). Convolutional neural networks (CNN) have gained vast popularity in computer vision tasks as well as many other domains. CNNs apply learnable convolutional filter to combine low-level features to build higher-level features. Tsinalis et al. (2016b) used a 1-dimensional variant of a CNN on the raw EEG data. Their method successfully includes both feature extraction and classification in one network (accuracy: 82%, F_1: 81%). Zhang and Wu (2017) use a complex valued CNN as a variant of the previous approach. The complex values make it possible to add more dimensionality to the decision boundary and are significantly speeding up the convergence of the learning process (accuracy : 87%, F_1: 82-93%). The most recent approach of Supratak et al. (2017) makes use of a combination of two CNNs for feature extraction, one with larger filters to capture slow oscillations and one with smaller filters to account for shorter recording events. The architecture is topped with two bidirectional LSTM layers and a skip-connection, and is fine-tuned end-to-end with a separate CNN pre-training. The results of this approach are similar to the other presented approaches (accuracy: 86%, F_1: 82%).

1.5 Introduction to Neural Networks

1.5.1 Artificial Neural Networks

ANNs are a class of machine learning techniques that are loosely based on computations made by biological neural networks. In ANNs' most basic shape, the Perceptron (Figure 2) consists of one neuron that takes a number of inputs in_k and multiplies each input by a weight w_k. It then sums up the products and uses an activation function such as a sigmoid to compute an output activation. So the output of one neuron is

$$O = sigmoid(\sum in_k * w_k)$$

The output activation (e.g. 0.76) is then compared to the desired output (e.g. 1.00) and an error term is calculated. For all input weights w_k, the gradient of the error term (loss function) can be calculated enabling us to change the weights in a direction that minimizes the loss. A learning rate is introduced to prevent overshooting the target and to ensure a slow approximation of a (locally) optimal solution.

$$\Delta w_k = (truth - output) * i_k * learningrate$$

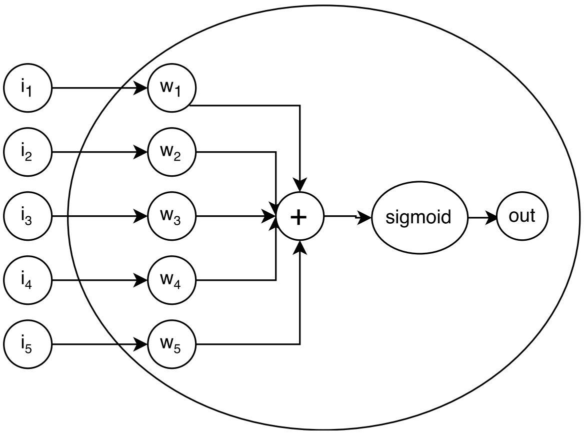


Figure 2: Schematic of a Perceptron

Several Perceptron neurons can be put in parallel to form a layer of neurons to allow for multi-class output which itself can be stacked to form a multilayer perceptron (MLP). An extension to the perceptron learning rule called error backpropagation is used to adjust the weights in a way similar as explained above: Each training example is passed through the network to compute an output. The difference between the output o_k and the ground truth y_k is used as an error term. The change necessary for each weight is now calculated using the derivative of the error with regard to the weight w_i (for details see Benvenuto and Piazza (1992). Figure 3 shows an MLP with 2 layers, a hidden layer h_k and an output layer o_k. The input layer i_k is classically not included in the layer count, as no computations are performed here. It has been shown that MLPs can theoretically approximate any mathematical function, given that the network architecture is complex enough (Hornik, 1991).

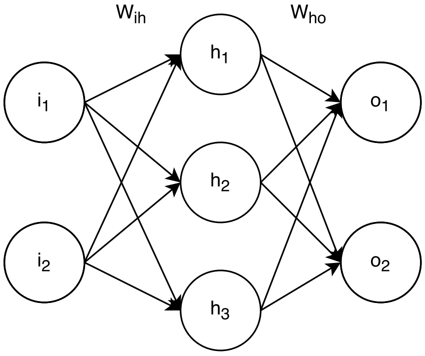
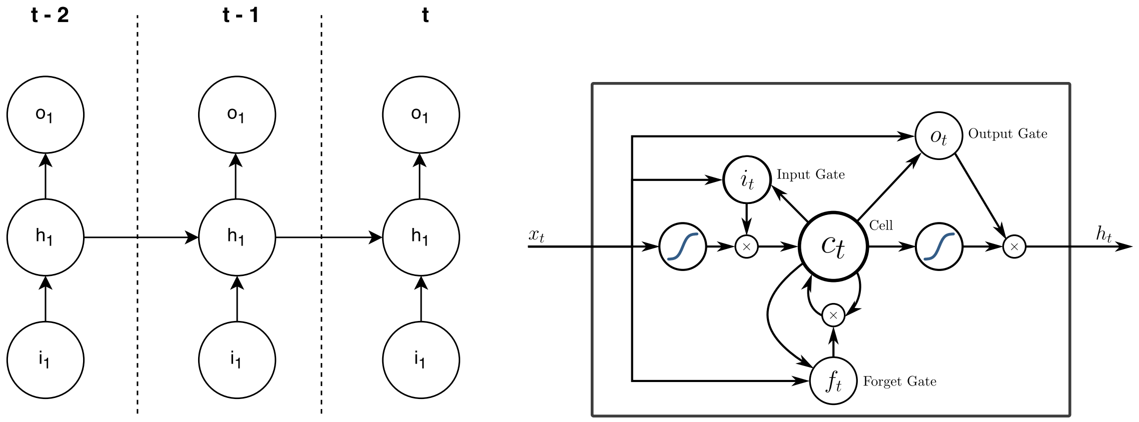


Figure 3: A simple MLP with one hidden layer. Each circle represents one computational node (i.e. Perceptron)

One problem when training an ANN is overfitting. As the network can learn any function mapping, it is possible to learn the training data by hard while losing the ability to generalize to unseen data. Therefore, the model always needs to be evaluated on a validation set to see if the validation error increases. Counter measures to overfitting need to be introduced such as limiting the weight magnitudes (L1 or L2 regularization (Girosi et al., 1995)) or enforcing the network to work in a noisy environment (Dropout regularization (Srivastava et al., 2014)). Many different modes of training an ANN are available. The learning process can be executed in batch-mode instead of per sample which means that weight changes are averaged and an update is only applied every couple of examples. The averaging of weights has the advantage of creating a smoother learning curve, as extreme samples do not stir the network in a wrong direction too much. Additionally, it is then possible to perform several calculations in parallel which speeds up the training process by orders of magnitudes and enables to make full use of the parallel processing power of the graphical processing unit (GPU).

1.5.2 Convolutional Neural Networks (CNN)

Many of the recent advances in the field of Artificial Intelligence were made possible by the use of so-called convolutional neural networks (CNNs) (Shea and Nash, 2015). A CNN in its simplest form comprises a convolutional kernel that is applied to an input signal where each entry of the kernel consists of a neural weight. With backpropagation, these weights can then be learned and adapt themselves to the features that are present in the input signal. Each kernel is 'scanning' over the input and returns the activation of that region for its filter kernel. Similar to the MLP, layers can be stacked, and each layer can build up a more abstract feature representation from features found in the previous layer. With this process, it is possible to extract abstract feature representations from raw data such as images or EEG signals.



(a) A simple RNN with one hidden neuron. At each timestep t the cell receives the regular input i_1 as well as the output from itself at $t-1$

(b) A LSTM cell with an input, output and forget gate plus a peephole connection that allows the gates to look at the current cell state.

Figure 4: Recurrent Neural Network Schematic[2]

1.5.3 Recurrent Neural Networks (RNN)

An RNN is a ANN variant where a cell does not only regard the current input but also its previous state (see Figure 4a for an illustration). One problem is that with each time-step, small bits of information are getting lost, making it hard for the network to keep longer time dependencies. LSTM neurons (Hochreiter and Schmidhuber, 1996) are an extension trying to solve this by introducing several gates that allow to explicitly read, write, and delete information from a neuron's memory. This way, information can be stored until needed and then deleted. Figure 4b shows a schematic of an LSTM cell. RNNs perform best whenever a temporal component is present and have been applied to many domains (Lipton et al., 2015).

[2] b) taken from `https://commons.wikimedia.org/wiki/File:Peephole_Long_Short-Term_Memory.svg`

1.5.4 Disadvantage of ANNs and Conclusion

The main disadvantage of ANNs are that it is not always possible to understand what features are learned by a network. Especially in higher layers, a network can learn abstractions that are hard to translate to a comprehensible representation. Additionally, ANNs need large amounts of data to generalize well. This makes their training time much higher than more classical algorithms such as random forests or support vector machines. Many hyper-parameters (network architecture, regularization, optimization algorithm, learning rate, ...) need to be optimized making expert knowledge necessary for the design choices. Nevertheless, ANNs are currently the most promising tool of artificial intelligence and machine learning. Almost all major breakthroughs in recent years have been accomplished using some form of ANN, and areas where their performance outrivals other machine learning approaches are endless: Super-human level object classification (He et al., 2016), super-human level Go playing (Silver et al., 2016), human level medical image analysis (Shen et al., 2017), speech recognition (Zhang et al., 2017) and production (van den Oord et al., 2016), machine translation (Wu et al., 2016) and many more (LeCun et al., 2015).

1.6 Aim of this Thesis

In this thesis, I want to look into the possibility of creating a robust automatic sleep stage classifier using artificial neural networks. Sleep EEG data is multimodal, complex and entails a temporal component ranging over many million data points (if recorded e.g. with 100 samples per second for several hours over a combination of multiple recording sites). Additionally, hand-crafting features to use for classification contains a subjective bias and therefore possibly limits the quality of classification. Therefore, an automatic feature extraction using a convolutional neural network (CNN) will be used. To capture the temporal dependency of sleep scoring, it will be explored if considering past epochs can improve performance by extending the network with a recurrent neural network using LSTM neurons. Additionally, it shall be explored if the use of EEG alone is sufficiently accurate for sleep stage classification or if adding EMG or EOG signals can improve the performance, especially the difficult-to-differentiate sleep states (e.g. Wake or S1). This is especially interesting for the future development of wearable or mobile sleep recording devices. All algorithms will be compared with a simple feed-forward neural network or an LSTM network using hand-crafted features as inputs. Different datasets will be used (see section 2.1), most prominently data from the National Sleep Research Resource (NSRR), a sample of over 500 participants. Although some automatic sleep scoring systems are commercially available, they are costly to obtain and obscure their algorithms (closed source). As there is most often no trial mode available, sleep researchers must buy the product before being able to assess its quality. This thesis should work as a basis towards the creation of an open-source program with an easy-to-use application programming interface (API) to be integratable in other EEG analysis software (e.g. SpiSOP and FieldTrip (Oostenveld et al., 2011)).

To explore the various setups the following research questions will be addressed:

1. Is automatic feature extraction able to perform similarly to using a hand-crafted feature approach?
2. Which channels aid classification? Is it possible to perform well with using less than the recommended three channels (EEG, EMG, EOG)?
3. How important is the temporal dependency for the classification?
4. How well does the classification quality transfer when tested on other data?

2 Methods

2.1 Sleep Datasets

For training and testing the following public dataset has been used:

(a) The Cleveland Children's Sleep and Health Study (CCSHS) dataset with full-night polysomnographic recordings from 515 participants (16-19 years) with minority population representation (Hibbs et al., 2014; Rosen et al., 2003; Spilsbury et al., 2005) from the National Sleep Research Resource website (Dean et al., 2016). Our expectation when choosing this dataset was to be able to train on homogeneous recordings from young, healthy participants in a broad population. Due to computational limitations only the first 50 recordings were included for most of this study.

For further testing, the following datasets were used:

(b) The public Sleep-EDF dataset (Sleep-EDFx)(Kemp et al., 2000) from the PhysioNet database (Goldberger et al., 2000) featuring 39 full-night recordings from 20 participants.

(c) The public St. Vincent dataset (UCD) with full-night recordings from 25 participants from the PhysioNet database (Goldberger et al., 2000).

(d) The private Episodic Memory and Sleep Assessment (EMSA) dataset (Wang et al., 2017) with 51 participant full-night polysomnographic recordings including 18 school children (8-12 years) and 33 adults (18-30 years) divided into the subsets EMSAch (children) and EMSAad (adults).

Figure 5 shows the distribution of sleep stages in the corresponding datasets plus their total number of epochs used in this study. All datasets except for the UCD feature a normal distribution of the sleep stages (Carskadon et al., 2005). The large amount of S1 in the UCD dataset might be due to the participant group being patients with breathing disorders (apnoe). Table 1 gives an overview of the descriptive characteristics of the datasets.

All datasets were scored in 30 second epochs. Most recordings included additional channels such as abdominal expansion, airflow, oximetry or ECG which have been disregarded in this study. As recommended by the AASM, the following electrodes have been used in the experiment: C3/A2 or FPz/Cz, EMG and EOG. The hypnogram annotations contain the following labels: Wake, Stage 1, Stage 2, Stage 3, Stage 4, REM as well as (Movement)

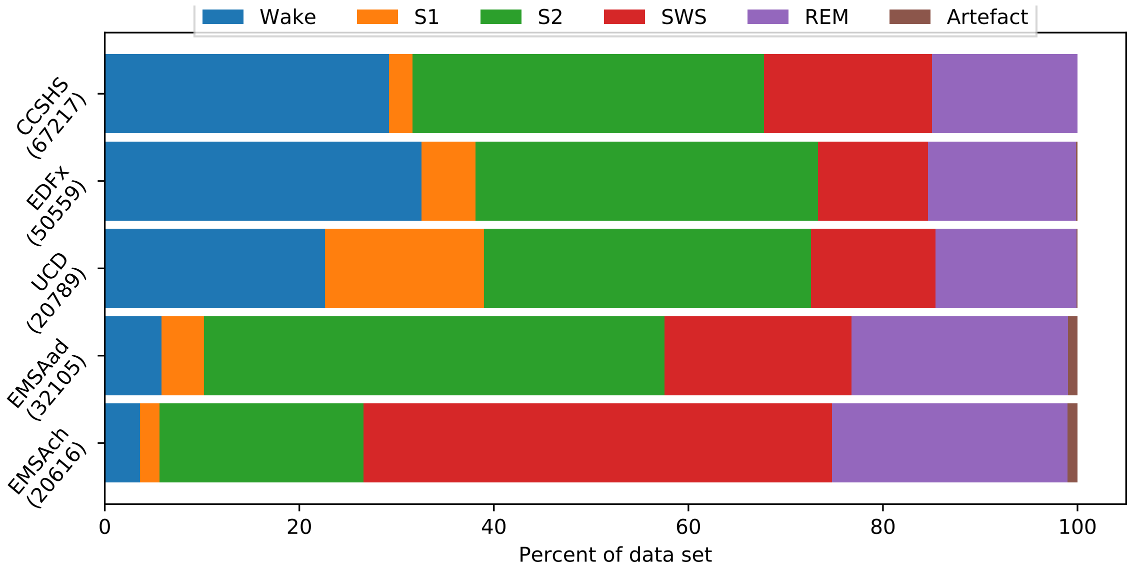


Figure 5: Distribution for sleep stages for the datasets. In brackets are the total numbers of epochs.

Artefact and Unclassified. Sleep stages 3 and 4 were combined to slow-wave sleep (SWS) as recommended by the new AASM sleep scoring guidelines (Iber et al., 2007). Movement artefacts were rare (see Figure 5) and have therefore been included in the 'Wake' label. Unclassified episodes were excluded. For facilitation of the data analysis and comparability between datasets, all signals were resampled at 100 Hz. This sampling frequency was chosen as it is the lower bound of sampling frequencies used by most commercial sleep datasets, while still including gamma frequencies up to 50 Hz. To remove electrode drift and artefacts the EEG and the EOG channel were high-pass filtered at 0.15 Hz while the EMG channel was high-pass filtered at 10 Hz. The data was then normalized by z-scoring all values over the entire dataset. Figure 6 shows a typical example of a 30 second epoch of the EEG during REM.

Table 1: Overview over datasets

Name	**CCSHS**	**Sleep-EDFx**	**UCD**	**EMSAad**	**EMSAch**
Participants	50 (515 total)	39	25	33	18
F/M	25/25	19/20	4/21	16/17	10/8
Age	16-19	21-101	28-68	18-30	8-12
Population	Healthy	Healthy	Obese and apnoea	Healthy	Healthy
Raters	N/A	Four raters	One rater	Two raters	Two raters
Rules	R&K rules	R&K rules	R&K rules	R&K rules	R&K rules
Epochs	67217	50559	20789	32105	20616
EEG Channel	C3-A2/ C4-A1	Fpz-Cz/ Pz-Oz	C3-A2/ C4-A1	C3-A2/ C4-A1	C3-A2/ C4-A1
Sample freq. EEG/EMG	128 Hz	100 Hz	128 Hz	100 Hz	100 Hz
Sample freq. EOG	50 Hz	100 Hz	64 Hz	100 Hz	100 Hz
Acquisition period	2006-2010	1987-1991	2002-2003	2012-2015	2012-2015

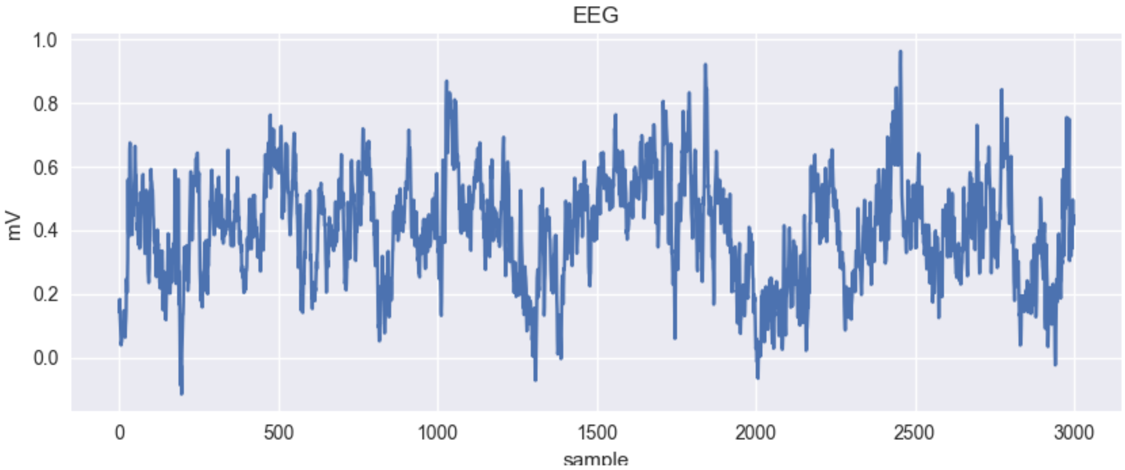


Figure 6: One epoch of REM sleep from the EEG recording

2.2 Feature Extraction: Hand-crafted vs. Automatic

Hand-crafted feature extraction is used in most of the current approaches (see 1.4) and contains features that are calculated on the raw EEG, EOG or EMG signal. Usually, this entails time-frequency transformations and clustering into common brain-wave frequencies (alpha, beta, gamma and delta). For the experiments reported below, 27 features were extracted as proposed by Längkvist et al. (2012) on 30 second segments of the raw data in addition to 10 other features. The proposed features consist of:

- 5 frequency bands (delta: 0.5-4 Hz, theta: 4-8 Hz, alpha:8-13 Hz, beta: 13-20 Hz, gamma: 20-50 Hz) for all channels plus an additional band covering the sleep spindle range (12-14 Hz)
- Mean, median and, variance for the EMG and EOG
- Minimum, maximum, position of minimum/maximum, and sign change of the EOG
- Kurtosis, entropy and, magnitude from all channels

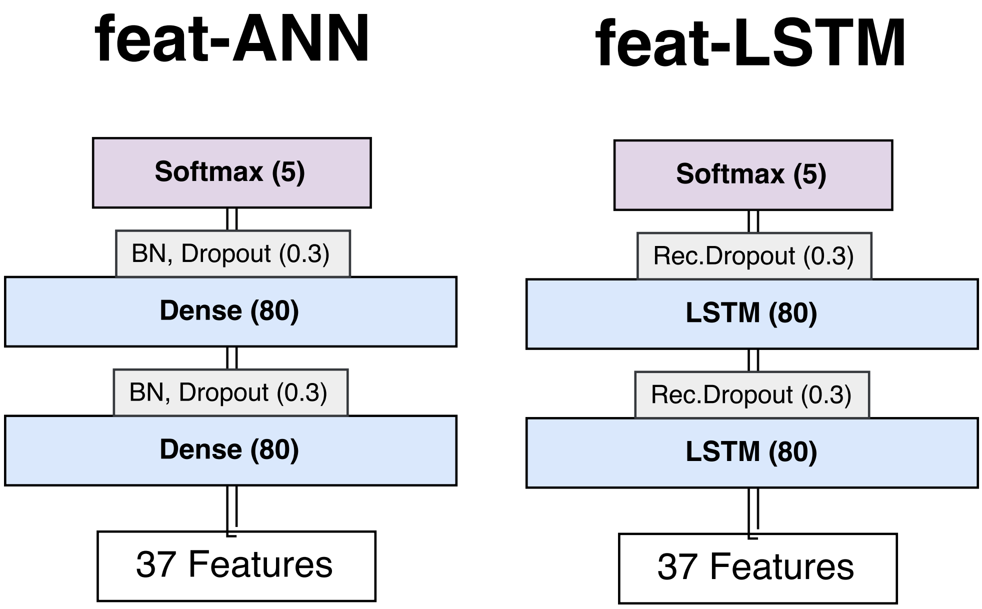
All features were z-scored before training.

2.3 Network Architecture and Training

A feed-forward ANN using the hand-crafted features (feat-ANN) as well as a CNN using the raw signals as input was used in the first part of this study. ReLU was used as an activation function for all layers while the final classification layer used a soft-max activation to allow a probabilistic output. A He-Norm initialization was used for the weights (He et al., 2016). Each computational layer was followed by a batch-normalization and a dropout operation. A full network description can be found in the appendix A.

The network architecture for the feat-ANN consisted of two layers of neurons with 80 neurons each and batch-normalization followed by a dropout with ratio of 0.3. These numbers had been chosen after a small hyper-parameter search using hyperopt (Bergstra et al., 2015) with 250 runs on the search-space: layers: 1-3, neurons: 5-100, dropout: 0.0-0.8. The architecture can be seen in Figure 7a).

The CNN architecture was chosen after a 5-fold cross validation on five different model architectures (larger and smaller filters, more and less layers). Due to the long runtime of the CNNs (2-3 hours per fold) and the very large parameter space, this model search could only be performed very coarsely. The winning model had a 1D convolutional layer with a



(a) feat-ANN

(b) feat-LSTM

Figure 7: Architectures for using hand-crafted features

filter size of 50 followed by two blocks of 1D convolution and max-pooling. The rationale behind this model was, that the filter size of 50 would enable the network to pick up the lower-bound frequency of 0.5 Hz which also represents the slow waves appearing in SWS. The following layers include more convolutions that should be able to create higher order features. The max-pool layers and the strides were added to reduce computational cost as well as to act as regularization. A L2-regularization with a λ-value of 0.05 has been applied to the first convolutional layer to prevent overfitting of the filters to noise present in the data, less rigid regularization was applied to the other convolutional layers (λ-value of 0.01). The CNN architecture scheme can be seen in Figure 8.

In the second part of this study, features of each sample were extracted from the neuron activations of the first fully-connected layer of the CNN model trained using all channels. The activations of fully-connected layers is generally seen as an abstract representation of the input signal. These are used as an input to a two-layer recurrent network using vanilla LSTM neurons to enable the network to consider previous epochs while classification (see Figure 8). Dropout as well as recurrent dropout is applied to the connections with a chance of 0.3. The LSTM network is trained on a sequence length of six epochs (three minutes), that means it sees six epochs sequentially and gives as an output the sleep stage of the last epoch. This parameter had been chosen after a search over sequences from two to ten and is in accordance with scoring rules that ask the scorer to look back up to five epochs in some cases. Similarly, the hand-crafted features are used to train a vanilla LSTM network (feat-LSTM). This network follows the structure of the feat-ANN and consists of two layers with 80 neurons each. Recurrent dropout of 0.3 had been applied to both layers. The same sequence length of six had been chosen. The network scheme can be seen in Figure 7b.

Each training run consisted of a 5-fold cross validation where 80% of the data are used for training and 20% are used for testing. Additionally a validation dataset is needed so that the 80% training data was furthermore split up again into 80% training data and 20% validation data to allow early-stopping. It was made sure that each participant record is only present in one split to prevent the bias of training and testing on the same participant. Training was performed using the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.0001 and stopped after no improvement could be seen on the validation set for 15 epochs. A batch-size of 512 was used and one epoch was defined as a run over the whole training set. Despite a large class imbalance, class balanced sampling with or without hard negative subsampling destabilized learning and was hence not applied. The training data had 3000 samples per epoch (30 seconds), of which a random crop of size 2800 was used per training step to enable a larger variance of input data (data augmentation). For validation and testing a centred crop has been used. The input of the network was therefore 2800 samples per channel (shape = 2800, 3 in case of all 3 channels). The cropped input of 28 seconds will most likely still contain almost all of the important characteristics of the epoch. For testing a mean prediction using two crops covering the whole epoch could be used as well to ensure correct classification. The Keras (Chollet, 2015) library for Python was used as an ANN training framework using Tensorflow (Abadi et al., 2015). Training was done using a GeForce GTX 1060 6GB or a cluster with two Tesla K40. All source code is available at `https://github.com/skjerns/AutoSleepScorer`.

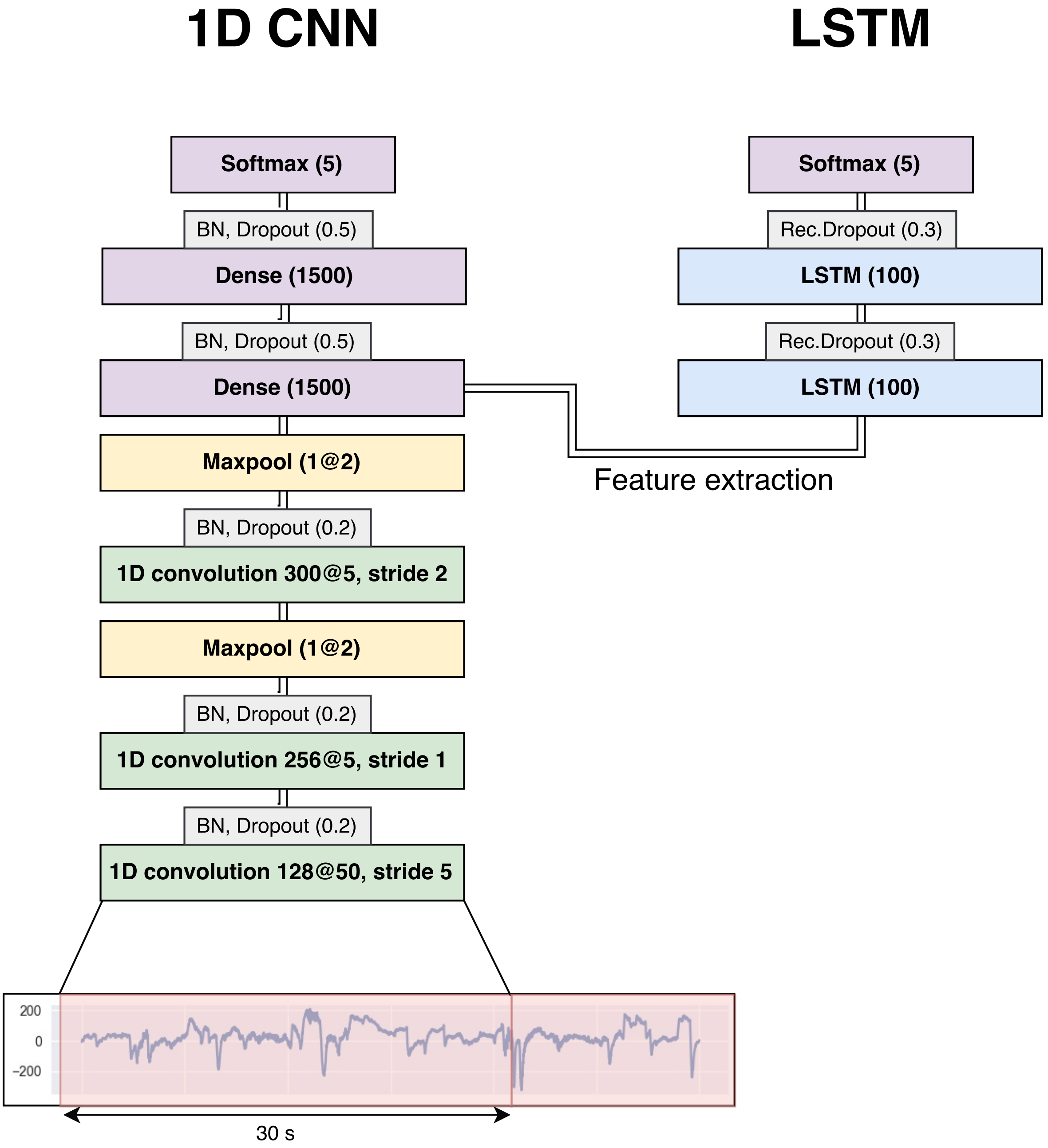


Figure 8: Diagram of the final CNN network (left) including the recurrent extension (right)

2.4 Performance Measures

To calculate the performance of a classifier, different scores can be used. Frequently, the accuracy is used as a measure which calculates the overall percentage of agreement between the prediction and the ground truth. The accuracy is sensitive to an imbalance in classes: As up to 40% of the classes are S2, we would get an accuracy of 40% just by classifying all epochs as S2 making the measure biased. Therefore, additional other measures are needed. Important terms in measuring a classifier performance in a binary case are the precision and the recall. The precision is the number of correctly identified positives divided by the total number of positive predictions. The recall is the number of correctly identified positives divided by the total number of positives in the ground truth. In the case of a multi-class problem, these scores can be calculated per class. To give an overall estimation of performance using these two measures, the F_1-score can be used. The F_1 score calculates the harmonic mean of the precision and the recall per class, which makes it insensitive to class imbalances. The overall F_1 score can be calculated as a harmonic mean between classes and is used as the primary performance measure in this study.

2.5 Design of Study

This study performed a set of experiments in which all conditions contained a comparison between the hand-crafted features (using feat-ANN and feat-LSTM) and the automatic feature extraction (using the CNN or CNN+LSTM). At first the performance per channel was assessed by training the networks with either only the EEG, or the EEG and one other channel (EOG or EMG), or all three channels (EEG, EMG and EOG). The EEG channel was always included as it is seen as vital in accordance with the AASM sleep scoring guidelines. In the next stage, it was analyzed if a temporal component can improve the classification performance by stacking the networks with LSTM neurons. It was assessed in which sleep stage the most confusion appears. For the initial experiments the first 50 recordings of the CCSHS dataset were used to reduce the otherwise long training times (7-10 hours run).

Based on these experiments, a final model and modality was chosen. Decisions were made based on the F_1-score on the test set. The winning model was then trained and evaluated further on the other datasets (other cohort) as well as participants 50-100 of the CCSHS (same cohort) to measure the performance of the network on other data. Transfer capabilities were assessed by training the network on the first 50 entries of the CCSHS and testing it on the other datasets.

3 Results

3.1 Channel Selection

The results of the comparison between different sets of channels can be found in Table 2. Shown are the mean accuracy and F_1-scores on the validation- and test data of the 5-fold cross validation with their corresponding range. Generally, it can be seen that adding more channels gives better results while using all channels yields the highest performance (CNN: val. F_1: 79.5%, test F_1: 77.0%, feat-ANN: val. F_1: 77.8%, test F_1: 76.7%). This increase when using more channels is more drastic for the feat-ANN. Using the EOG as a second channel seems to give equally good results as using all channels for the CNN. The mean confusion matrices of the cross-validation folds using all channels can be seen in Figure 10a for the CNN and Figure 10b for the feat-ANN. The results indicate that the feat-ANN is not good at detecting S1 while the CNN is able to correctly identify it in 50% of the cases. Additionally, differences of confusion matrices have been created. They illustrate the performance gain when using all channels compared to using only a subset of channels (see Figure 11). S1 is the stage that profits most from including more channels while there is almost no difference between EEG+EOG and using all channels except for S1 recognition. Figure 9 shows a selection of features that the first layer of the CNN learned. Figure 9 bottom shows filters that were learned by a network without L2-regularization.

Table 2: Results for the channel selection: Mean and range of the cross validation. Best results per row in bold.

CNN	**EEG**	**EEG+EMG**	**EEG+EOG**	**All**
Validation Acc.	85.4% (82.4-88.1)	85.7% (84.4-86.8)	86.3% (83.3-89.3)	**86.8%** (83.5-90.5)
Validation F_1	75.8% (73.4-78.4)	77.4% (75.1-79.6)	78.8% (76.8-82.2)	**79.5%** (75.6-83.4)
Test Acc.	84.7% (81.1-88.3)	83.7% (79.6-85.5)	**85.8%** (82.1-89.4)	85.7% (81.7-87.1)
Test F_1	74.6% (72.8-77.8)	74.5% (72.4-76.6)	76.9% (74.5-80.5)	**77.0%** (74.1-78.6)
feat-ANN				
Validation Acc.	82.4% (79.9-83.8)	83.9% (82.3-85.1)	87.0% (85.1-88.5)	**87.9%** (86.8-89.3)
Validation F_1	66.4% (64.5-67.7)	71.3% (70.2-71.8)	74.5% (72.0-76.6)	**77.8%** (76.1-79.1)
Test Acc.	82.2% (80.2-83.4)	83.6% (81.8-86.2)	86.2% (84.1-87.4)	**86.7%** (84.2-87.7)
Test F_1	65.8% (64.4-66.4)	71.7% (67.8-75.2)	72.6% (71.6-74.8)	**76.7%** (74.9-78.5)

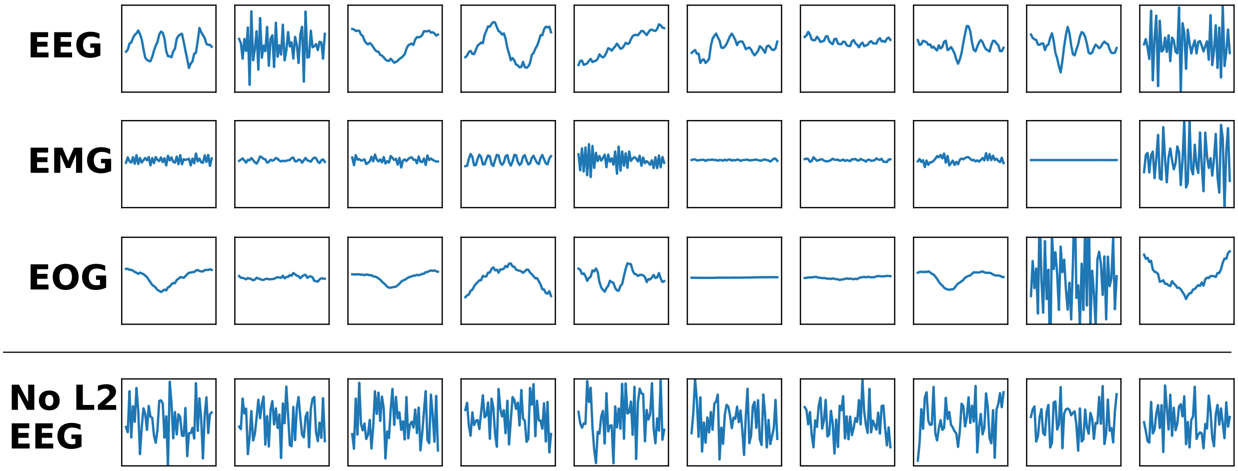
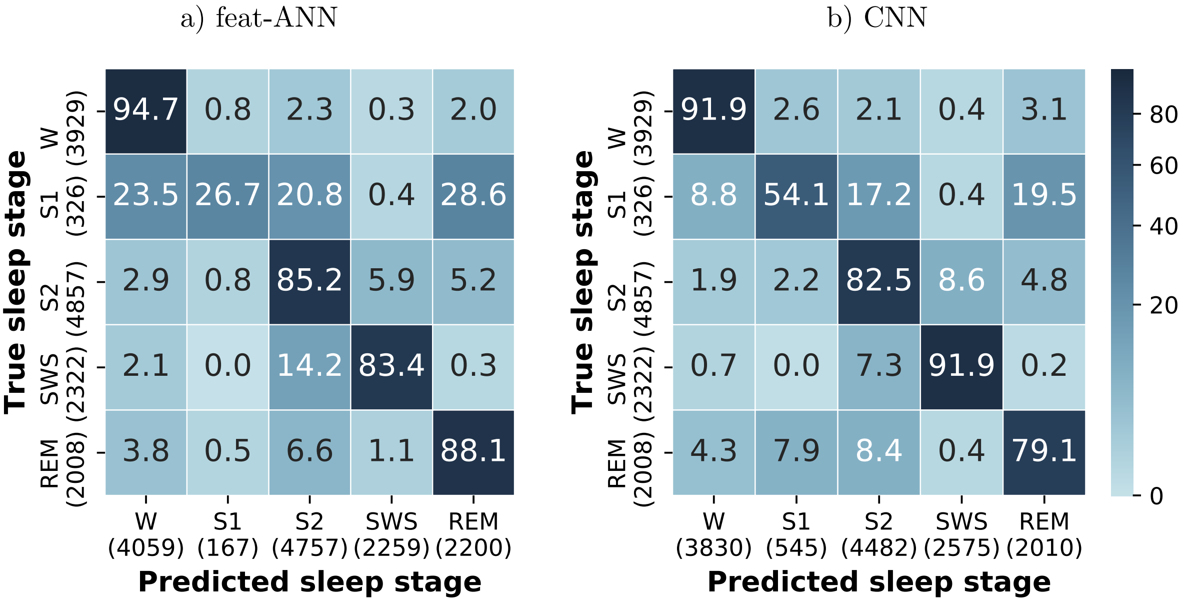
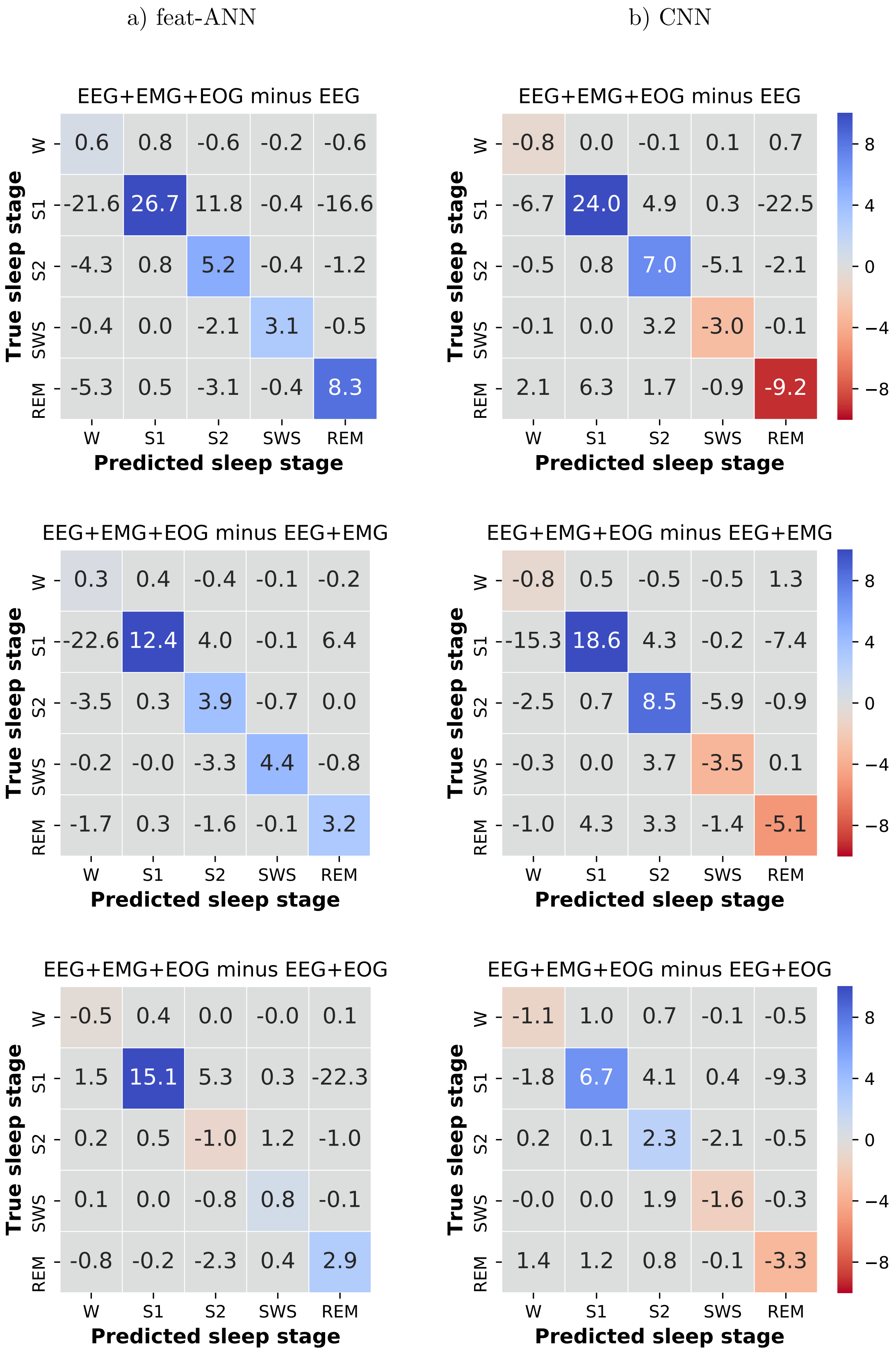


Figure 9: Top: Visualization of 10 of the 128 filters of the first CNN layer for each channel. Bottom: 10 filters of the EEG without L2-regularization. Each filter covers 50 milliseconds


a) feat-ANN

b) CNN

Figure 10: Confusion matrices for using all channels



a) feat-ANN

b) CNN

EEG+EMG+EOG minus EEG (feat-ANN)

	W	S1	S2	SWS	REM
W	0.6	0.8	-0.6	-0.2	-0.6
S1	-21.6	26.7	11.8	-0.4	-16.6
S2	-4.3	0.8	5.2	-0.4	-1.2
SWS	-0.4	0.0	-2.1	3.1	-0.5
REM	-5.3	0.5	-3.1	-0.4	8.3

EEG+EMG+EOG minus EEG (CNN)

	W	S1	S2	SWS	REM
W	-0.8	0.0	-0.1	0.1	0.7
S1	-6.7	24.0	4.9	0.3	-22.5
S2	-0.5	0.8	7.0	-5.1	-2.1
SWS	-0.1	0.0	3.2	-3.0	-0.1
REM	2.1	6.3	1.7	-0.9	-9.2

EEG+EMG+EOG minus EEG+EMG (feat-ANN)

	W	S1	S2	SWS	REM
W	0.3	0.4	-0.4	-0.1	-0.2
S1	-22.6	12.4	4.0	-0.1	6.4
S2	-3.5	0.3	3.9	-0.7	0.0
SWS	-0.2	-0.0	-3.3	4.4	-0.8
REM	-1.7	0.3	-1.6	-0.1	3.2

EEG+EMG+EOG minus EEG+EMG (CNN)

	W	S1	S2	SWS	REM
W	-0.8	0.5	-0.5	-0.5	1.3
S1	-15.3	18.6	4.3	-0.2	-7.4
S2	-2.5	0.7	8.5	-5.9	-0.9
SWS	-0.3	0.0	3.7	-3.5	0.1
REM	-1.0	4.3	3.3	-1.4	-5.1

EEG+EMG+EOG minus EEG+EOG (feat-ANN)

	W	S1	S2	SWS	REM
W	-0.5	0.4	0.0	-0.0	0.1
S1	1.5	15.1	5.3	0.3	-22.3
S2	0.2	0.5	-1.0	1.2	-1.0
SWS	0.1	0.0	-0.8	0.8	-0.1
REM	-0.8	-0.2	-2.3	0.4	2.9

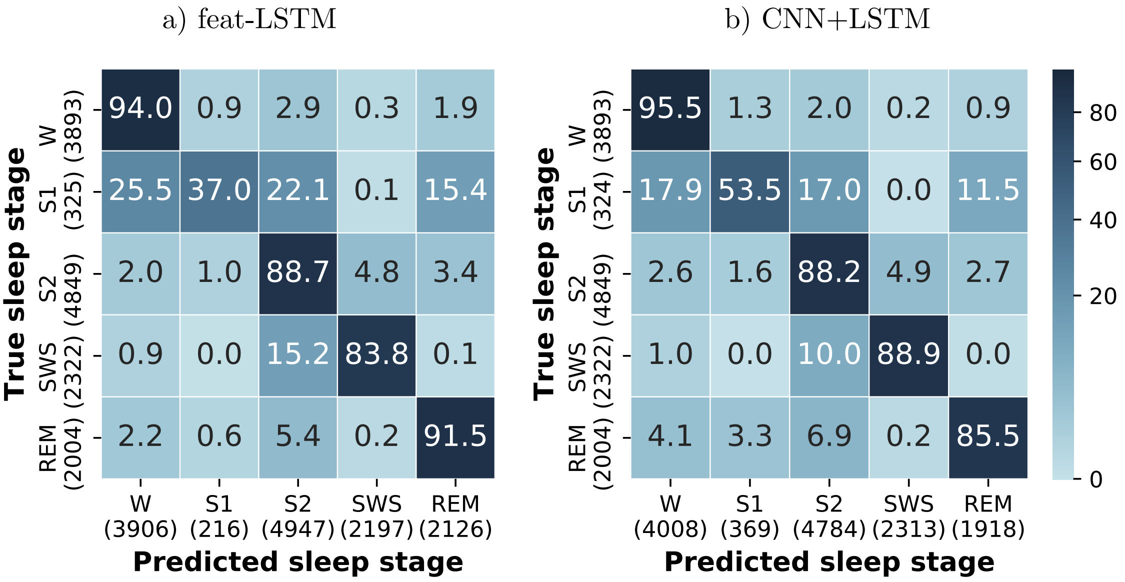
EEG+EMG+EOG minus EEG+EOG (CNN)

	W	S1	S2	SWS	REM
W	-1.1	1.0	0.7	-0.1	-0.5
S1	-1.8	6.7	4.1	0.4	-9.3
S2	0.2	0.1	2.3	-2.1	-0.5
SWS	-0.0	0.0	1.9	-1.6	-0.3
REM	1.4	1.2	0.8	-0.1	-3.3

Figure 11: Differences of confusion matrices for using all channels vs. using a subset (only the diagonal is colour coded), left: feat-ANN, right: CNN

3.2 Temporal Dependency

Table 3 shows the results for the feat-LSTM with hand-crafted features shows the results for the hand-crafted feature approach using the feat-LSTM network and the results for the CNN+LSTM network with features extracted from the CNN of the previous section. Results are shown using all three channels. The test performance of the CNN+LSTM approach (F_1: 81.4) was marginally better than the using the hand-crafted features (F_1: 80.2). The confusion matrices and the difference matrix can be seen in Figure 12. It can be seen that S1 was correctly classified in 53.5% of the cases with the CNN+LSTM whereas only 37.0% of S1 was recognized by the feat-LSTM. This difference in performance was also reflected in the recognition of REM sleep where the feat-LSTM was around 7%-points better than the CNN+LSTM. Figure 13 shows the predictions of the network in contrast to the scoring. Some brief state changes (arousals) seem to not be captured by the CNN+LSTM. A Pearson correlation revealed a significant correlation between the annotated state changes per participant and the F_1-scores (p=0.03) but not with the accuracies (p=0.92).


a) feat-LSTM

b) CNN+LSTM

Figure 12: Confusion matrices for the recurrent extension

Table 3: Results for the feat-LSTM and the CNN+LSTM approach.

	feat-LSTM	**CNN+LSTM**
Validation Acc.	88.9% (87.6-89.8)	**90.5%** (89.6-91.7)
Validation F_1	80.7% (78.5-82.1)	**84.1%** (82.0-86.2)
Test Acc.	88.6% (86.7-89.6)	**89.2%** (87.8-90.3)
Test F_1	80.2% (78.9-81.5)	**81.9%** (80.4-83.3)

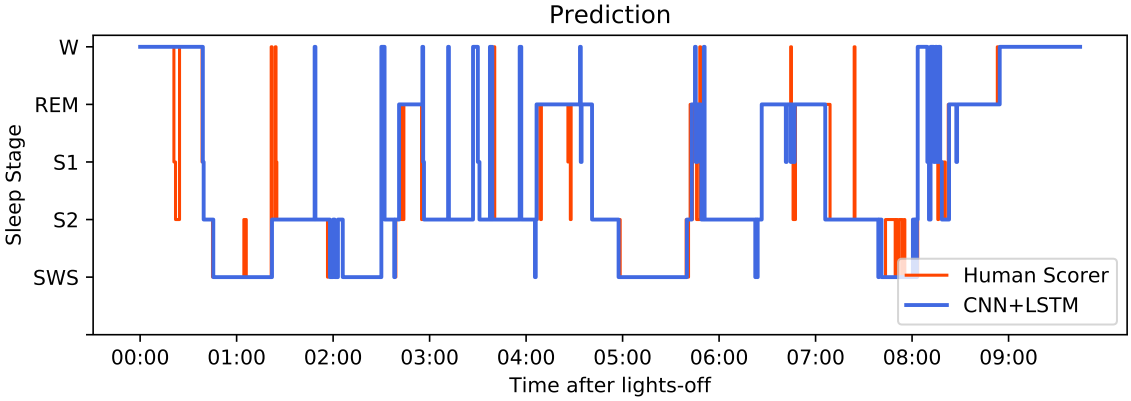


Figure 13: Hypnogram of stage predictions of the CNN+LSTM on one participant of the CCSHS dataset

3.3 Using different datasets

A number of public datasets are available that are generally used in the training and testing of sleep stage classifiers. To make a fair comparison with other approaches, classifiers performance is only valid if tested on the same dataset. Table 4 shows an overview of results when training and testing the CNN+LSTM and feat-LSTM networks on the previously mentioned datasets (Sleep-EDFx, UCD, EMSA). It can be seen that both the results on the Sleep-EDFx as well as the EMSAad are comparable to the result on the CCSHS (Table 3). On the UCD dataset, performance was worse, mostly due to the poor recognition of S2. The feat-LSTM showed consistently worse performance with poor performance on the UCD dataset and the EMSAch. Confusion matrices for the experiments can be found in Figure 14.

Table 4: Performance on different datasets

CNN+LSTM	**Sleep-EDFx**	**UCD**	**EMSAad**	**EMSAch**
Test Acc.	87.0% (83.3-90.0)	73.1% (71.3-75.4)	87.2% (85.3-88.4)	73.8% (68.7-78.5)
Test F_1	79.8% (76.2-82.8)	71.5% (66.6-74.0)	77.2% (75.1-79.1)	62.3% (54.8-68.0)
feat-LSTM				
Test Acc.	76.2% (71.1-82.6)	62.5% (58.6-67.0)	80.4% (77.7-83.7)	71.4% (67.6-75.0)
Test F_1	66.0% (59.8-74.8)	57.1% (54.8-59.5)	64.1% (60.2-67.9)	56.6% (49.4-60.9)

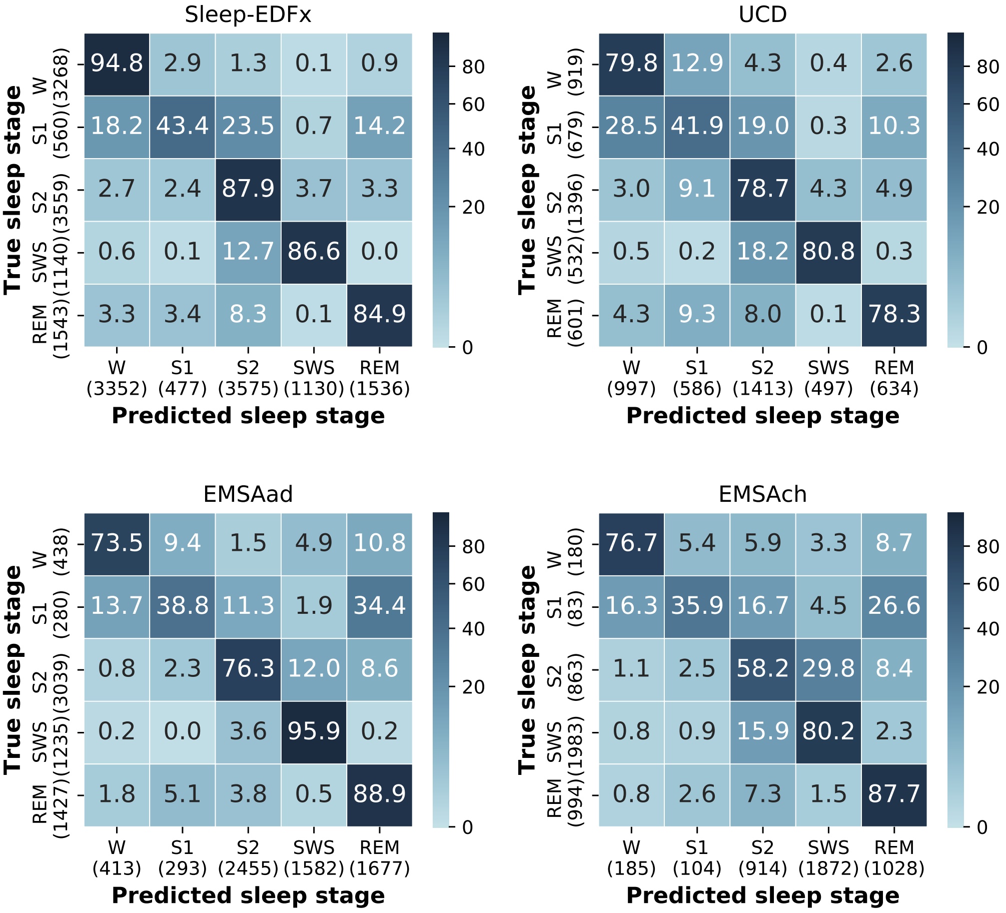


Figure 14: Confusion matrices of the CNN+LSTM for training and testing on other datasets

3.4 Transfer performance

Table 5 shows results[1] of the classifier trained on the first 50 participants (with 20% of participants used for early stopping) of the CCSHS dataset (CCSHS50) and tested on the other datasets as well as participant 50-100 of the CCSHS dataset (CCSHS100). The datasets have been z-mapped, that means normalization values are taken from the CCSHS50 dataset and a z-scoring is applied to the new dataset using these values. It can be seen, that results are generally high but fail to transfer in some cases. As expected the transfer to records from the same dataset is very high, even higher than for the first 50 records. The CNN+LSTM network seems to totally fail to transfer to the UCD when z-mapped, while results on the EMSAch are poor for both classifiers. Transfer performance of S1 detection seem to be consistently low around 30%. Confusion matrices can be found in Figure 15.

[1] A problem with the scaling of the UCD dataset was found making the classifier only predict REM when z-scoring. The second scores are provided with normal z-scoring on the same dataset.

Table 5: Transfer results when trained on the CCSHS50

CNN+LSTM	**CCSHS100**	**Sleep-EDFx**	**UCD**	**EMSAad**	**EMSAch**
Test Acc.	90.6%	80.8%	14.6/61.4% [3]	82.6%	65.0%
Test F_1	83.5%	71.6%	5.1/51.6%[3]	72.3%	56.8%
feat-LSTM					
Test Acc.	91.0%	73.3%	62.5%	75.4%	59.6%
Test F_1	83.2%	61.2%	55.9%	60.8%	48.5%

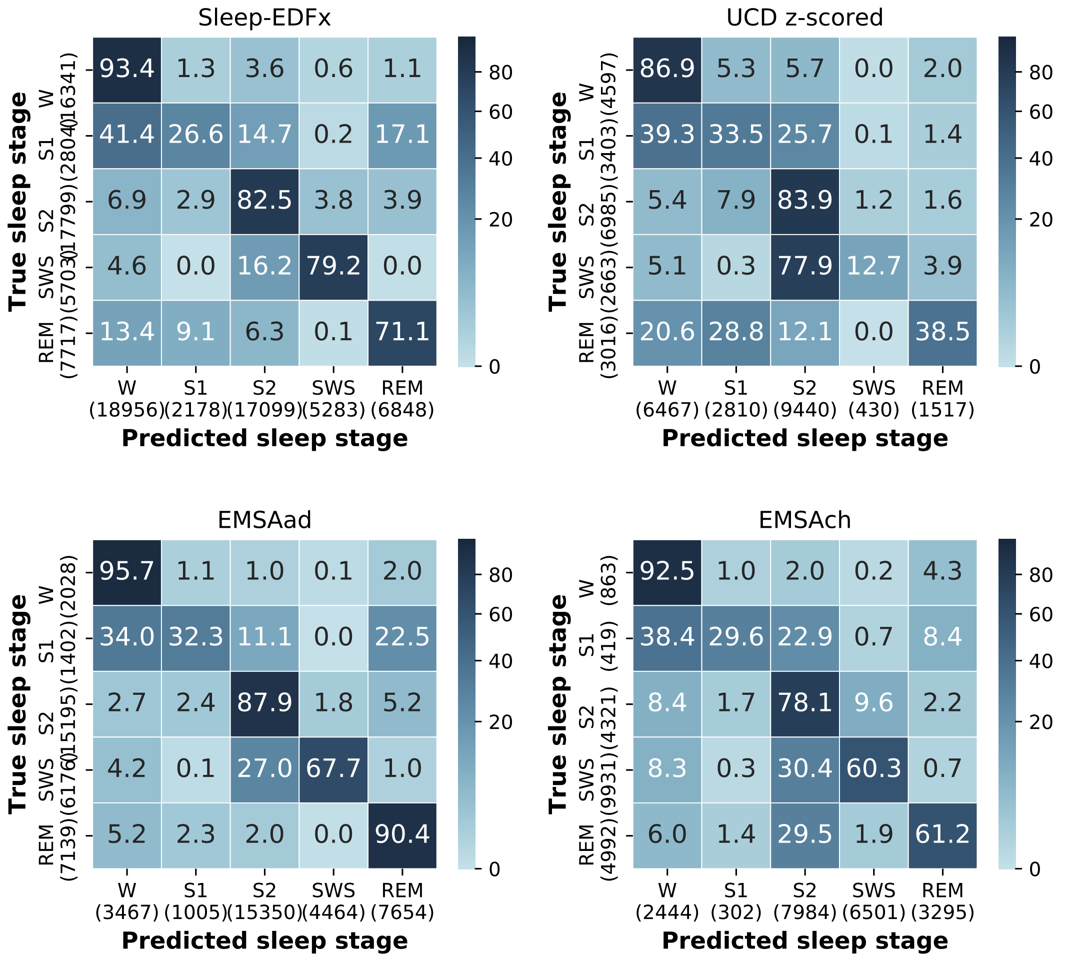


Figure 15: Confusion matrices of the CNN+LSTM for training on the CSHS50 and testing on other datasets

[3] A problem with the scaling of the UCD dataset was found so that the z-mapping failed. The second scores are provided with an another optimal scaling

4 Discussion

This thesis introduces a convolutional neural network as an automatic feature extraction method to train a recurrent neural network for the classification of sleep stages. The results show that a classifier trained on all channels as well as access to previous epochs performs best and reaches similar results as other studies on automatic feature extraction (see Table 3). The classifier shows similar performance to human raters, as reflected by the inter-rater agreement rates per sleep stage (Wake: 68-89%, S1: 23-74%, S2: 79-90%, SWS: 45-80%, REM: 78-94% (Silber et al., 2007)). Furthermore, using only the EEG and the EOG seems to be sufficient as also reported in Supratak et al. (2017). The comparison with a hand-crafted feature approach shows that the CNN+LSTM performs similarly well but shows superior performance in the recognition of S1.

4.1 Channel Selection

We can see that the feat-ANN profited most from using all channels, while the CNN seems to have equally good results in the EEG+EOG as with all channels. One reason for this could be that the CNN is able to extract latent features from channels, for instance it could infer the muscle tonus from artefacts in the EEG or EOG, making the inclusion of EMG superfluous. In the hand-crafted case, some features are only extracted from specific channels such as zero-crossings to detect eye-movements. This could explain the increase in performance when using all channels for the feat-ANN. This result is also reflected in Supratak et al. (2017) who are using the EEG channel with the EOG as a reference with a similarly high performance.

The confusion matrices show that correct recognition of S1 is mainly responsible for improvements. The CNN is able to obtain a reasonably high recognition of S1 (around 50%) which is close to the inter-rater reliability for that phase (23-74% (Silber et al., 2007) and even 18-42% for intra-rater reliability (Whitney et al., 1998)). The recognition of S1 is a well know problem reported in many other publications (Hassan and Bhuiyan, 2017). Given these facts, it is not surprising that the feat-ANN is even worse at recognizing S1 (27%). For the CNN, the improvements per class are not so clear. While some sleep stages (S1, S2) clearly profit from using more channels, there seems to be a trade-off at recognizing REM. This behaviour is surprising given that the classifier has access to the same and more information. In certain conditions, such as home-settings, it might be necessary to use as few channels as possible. These results show that a combination of EEG and EOG is already able to deliver satisfactory results. While in a clinical set-up all channels will be available, this indicates that for future mobile sleep recording devices a set-up of two electrodes could suffice.

The feature maps (Figure 9) also illustrate that the features learned by the network are close to events described by the AASM guidelines. In the EEG up-phases of a slow-oscillation, sleep-spindles as well as K-complexes can be observed, while rolling or sharper eye-movements can be detected in the EOG. The EMG shows different muscle tone activations. This is similar to what other studies performing automatic feature extraction are reporting (Längkvist et al., 2012; Supratak et al., 2017; Zhang et al., 2016; Zhang and Wu, 2017). For all three channels, there was at least one feature learned representing recording noise or artefacts as given by a muscle movement. The visualization also demonstrates that the L2-regularization is necesarry to prevented overfitting to noise, which is also reported by Supratak et al. (2017). While the network without L2-regularization was still having similar performance on the validation set, this might pose a problem when being used on other datasets with different noise patterns.

4.2 Temporal Dependency

The temporal extension shows a clear improvement in overall performance as well as in most individual sleep stages for both approaches. The CNN has maintained the relatively high recognition of S1 (54%), while the feat-LSTM could gain performance for that class (37%). The automatic approach slightly outperforms the hand-crafted approach except for stage REM (Figure 12c). Overall the results show that S1 remains the biggest problem for the classifier followed by REM. Other researchers successfully applied over- or under-sampling to tackle this problem (Supratak et al., 2017; Tsinalis et al., 2016b), where S1 epochs are duplicated or other stages reduced so that batches have the same number of samples per of stages. Experimenting with this technique showed a boost of S1 recognition to 70%, however, this also reduced recognition of other stages. The importance of S1 in a clinical setting is questionable as only few epochs contain S1 (∼3-5%) and its recognition by experts exhibits the highest variability. Therefore, a correct classification of S1 at the expenses of other stage recognition seems not advisable. Furthermore, an uncertainty-label could aid clinicians in the manual correction of S1 epochs to their subjective preference. In general, most confusion appeared between sleep stages that have similar features such as S2 and SWS or S1 and Wake. Figure 13 depicts that some errors are made when there are short stage-changes annotated, especially brief awakenings which were confirmed by the correlation analysis of state changes and prediction performance. It might be possible to use a second stage model for arousal detection to correct these problems.

Table 6: Comparison with other publications

Study	**Dataset**	**Method**	**Accuracy**	F_1
Längkvist et al. (2012)	UCD	DBN+HMM	67%	65%
Zhang et al. (2016)	UCD	SDBN+SVM+KNN+HMM	91%	89% [4]
Zhang and Wu (2017)	UCD	Complex valued CNN	87%	82% [4]
This study	UCD	CNN+LSTM	73%	72%
Tsinalis et al. (2016b)	Sleep-EDFx	CNN	81%	74%
Supratak et al. (2017)	Sleep-EDFx	CNN + bidirectional LSTM	82%	77%
This study	Sleep-EDFx	CNN+LSTM	87%	80%
This study	CCSHS	CNN+LSTM	89%	81%

4.3 Performance on Other Datasets

The calculation on other datasets shows that the artificial neural network is also able to perform well when using data that was not used in the development and fine-tuning of the model. Table 6 shows a comparison with other approaches that applied automatic feature extraction methods: Results on the Sleep-EDFx dataset show that the CNN+LSTM is able to achieve state-of-the art performance. While this study used information from all three channels, an important difference is that the results from Tsinalis Tsinalis et al. (2016b) and Supratak et al. (2017) focus on a single channel (EEG in Tsinalis et al. (2016b), EEG with EOG as reference in Supratak et al. (2017) approach. The results from the UCD perform significantly worse than the two studies by Zhang et al. (2016, 2017). Zhang et al. (2016), however, only used a sub-set of 10 recordings with sleep efficiency higher than 82% and both (Zhang et al., 2016; Zhang and Wu, 2017) were using a one-against-all classification instead of a 5-class problem classification making it again hard to compare results. In general, it is worth discussing how valuable the UCD dataset for classifier evaluation is as it features an abnormal distribution of sleep stages (see dataset description section 2.1).

4.4 Transfer Performance

To estimate a realistic performance of a real-world system it is important to test the transfer performance of the classifier on data recorded under different conditions. Unfortunately, reports of this measure are rare, none of the referenced papers in this study are assessing it.

The results (Table 5) show that the network is able to generalize well in most cases. Best results were obtained when using recordings from the same database. This is not surprising

[4] F_1 scores are calculated averages of scores per class of the one-vs-all classification as Zhang et al. do not provide those scores themselves

as all factors such as scoring-style, participant cohort, as well as recording modalities have been the same. Performance was high (accuracy 83%, F_1 72%) for a similar cohort (young adults) using the same set of channels. Surprising is the high transfer performance on the Sleep-EDFx (accuracy 81%, F_1 72%) as recordings were made using different EEG channels (Fpz-Cz) as well as a much broader age range (21-101). This shows that the learned features are general enough to be recognized in older participants using other channels. Transfer rates to data obtained from children were not as high (accuracy 65%,F_1 57%). One reason for this might be the differences in EEG activity and sleep stage distribution of children (Feinberg and Campbell, 2010), however, results obtained from training and testing only on the EMSAch show that the network is generally not good in this cohort. As major brain restructuring is happening within the age group it could be possible that no coherent scoring is possible to begin with. No study is known to the author assessing the reliability of scoring children sleep EEG (Feinberg and Campbell, 2010; Campbell and Feinberg, 2009). The absolute failure of the algorithm on the UCD dataset while using z-mapping could be traced back to an abnormal scaling of the data-files and could only be reverted by using a normal z-scoring. Even with the new scaling the performance was poor (accuracy 61.4, F_1 51.6). Surprisingly the feat-LSTM did not suffer from this scaling issue but also only showed an accuracy of around 63% (F_1 56%). The UCD features pathological sleep of apnoe patients which might explain the poor transfer. All other datasets profited from using z-mapping instead of per-dataset z-scoring on the whole dataset or z-scoring per recording. Furthermore, while performing a parameter search for an optimal scaling parameter, it was found out that there are certain dataset-dependent parameters that are beneficial for performance transfer that cannot be easily derived from the datasets. Further research needs to be done to find a way of consistently normalizing the input signal to prevent such a problem.

Two possibilities can be imagined to obtain a higher generalizability of the algorithm: First, it could be tried to train the classifier with a larger variety of subjects and recording modalities. It is questionable, however, if such an all-in-one classifier is able to capture the large variability present throughout human sleep recordings. Another approach would be to train specific classifiers for sub-groups. One obstacle for this case is that among the available public datasets only few groups are present and datasets for pathological cases (besides apnoe) are rare.

4.5 The Future of Automatic Sleep Scoring

This study, amongst others (Supratak et al., 2017; Tsinalis et al., 2016b; Zhang et al., 2016; Zhang and Wu, 2017) shows that automatic sleep scoring is able to be carried out by deep neural networks and reaches state-of-the-art results. However, in recent years most proposed automatic sleep scoring systems are reaching similar performance levels and it is questionable if improving those is possible and should be the goal of future studies. It is likely that the current limits are set by the inconsistencies of the labels and not the capabilities of the classifiers. Despite the widespread application of sleep scoring, there are only few studies trying to address the problem of the inter-rater reliability (Silber et al., 2007) and only one study is known to the author to report the intra-rater reliability (Whitney et al., 1998). More research in this direction is needed to know if significant performance improvements are possible at all.

For an application in a clinical setting the following points should be addressed by future studies:

(a) None of the other studies report the transfer-capabilities on different datasets. This performance measure is far more important than validation results on the same dataset and gives a sense of what clinicians and researchers can expect when using the system on their own data.

(b) Despite the multitude of publications, on the subject only a few of them get integrated into user-level systems. It is therefore not surprising to see most clinicians and researchers not using automatic sleep stage scoring, simply because of a lack of possibilities. The focus should therefore be shifted on producing usable products. For this it might help to publish results in an open-source manner.

Besides these issues, it should be discussed to what extent the current scoring standard should be altered to allow for a higher consistency in scoring (see Section 1.3). Machine learning on huge datasets could aid in the process of finding new definitions, for instance using unsupervised deep learning methods. A data-driven, clear definition of sleep stages would enable sleep research to be more accurate and might decrease or even eliminate the bias of the human rater.

5 Conclusion

This thesis combines a convolutional neural network with a recurrent long-short term memory network for the classification of sleep stages. The networks were able to reach state of the art classification performance (accuracy: 89%, F_1-score: 81%) on the public CCSHS dataset. Transfer performance without re-training to other datasets of a similar cohort was high (accuracy 81-91%, F_1-score: 72-84%). It was demonstrated that automatic feature extraction using deep learning is possible and that the CNN learns filters similar to those described by the AASM. It was shown that the network performs best with all three channel modalities namely EEG, EOG, and EMG, but a combination of EEG and EOG might be sufficient. Wearable devices could therefore acquire good results by only using two electrodes. There seems to be a ceiling effect in the classification performance of automatic sleep scoring systems given by the inconsistency of human scoring. Furthermore, automatic sleep scoring is still not established in the field despite the clear advantages of having a higher consistency while using the same classifier. The future of automatic sleep stage classification should therefore focus on creating usable systems rather than on further improving classification accuracy as compared to human raters. It should be discussed to what extent the current scoring standard should be revised on the basis of a data-driven methodology.

The source code of the thesis is available at `https://github.com/skjerns/AutoSleepScorer`

References

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Man, R. Monga, S. Moore, D. Murray, J. Shlens, B. Steiner, I. Sutskever, P. Tucker, V. Vanhoucke, V. Vasudevan, O. Vinyals, P. Warden, M. Wicke, Y. Yu, and X. Zheng (2015): "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *White Paper*, 1, 19.

Barkoukis, T. J. and A. Y. Avidan (2007): *Review of Sleep Medicine*, Philadelphia, PA: Elsevier, fourth edi ed.

Benvenuto, N. and F. Piazza (1992): "The backpropagation algorithm," *IEEE Transactions on Signal Processing*, 40, 967–969.

Bergstra, J., B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox (2015): "Hyperopt: a Python library for model selection and hyperparameter optimization," *Computational Science & Discovery*, 8.

Campbell, I. G. and I. Feinberg (2009): "Longitudinal trajectories of non-rapid eye movement delta and theta EEG as indicators of adolescent brain maturation." *Proceedings of the National Academy of Sciences of the United States of America*, 106, 5177–80.

Campbell, S. S. and I. Tobler (1984): "Animal sleep: A review of sleep duration across phylogeny," *Neuroscience and Biobehavioral Reviews*, 8, 269–300.

Carskadon, M. A., W. C. Dement, and Others (2005): "Normal human sleep: an overview," *Principles and practice of sleep medicine*, 4, 13–23.

Chollet, F. (2015): "Keras: Deep Learning library for Theano and TensorFlow," *GitHub Repository*, 1–21.

Cirelli, C. and G. Tononi (2008): "Is sleep essential?" *PLoS biology*, 6, e216.

Danker-Hopfe, H., P. Anderer, J. Zeitlhofer, M. Boeck, H. Dorn, G. Gruber, E. Heller, E. Loretz, D. Moser, S. Parapatics, B. Saletu, A. Schmidt, and G. Dorffner (2009): "Interrater reliability for sleep scoring according to the Rechtschaffen & Kales and the new AASM standard," *Journal of Sleep Research*, 18, 74–84.

Dean, D., A. Goldberger, R. Mueller, M. Kim, M. Rueschman, D. Mobley, S. Sahoo, C. Jayapandian, L. Cui, M. Morrical, S. Curovec, G. Zhang, and S. Redline (2016): "Scaling Up Scientific Discovery in Sleep Medicine: The National Sleep Research Resource," *Sleep*, 39, 1151–1164.

Ermis, U., K. Krakow, and U. Voss (2010): "Arousal thresholds during human tonic and phasic REM sleep: Phasic and tonic REM sleep," *Journal of Sleep Research*, 19, 400–406.

Feinberg, I. and I. G. Campbell (2010): "Sleep EEG changes during adolescence: An index of a fundamental brain reorganization," *Brain and Cognition*, 72, 56–65.

Franzen, P. L. and D. J. Buysse (2017): "Sleep in psychiatric disorders," in *Sleep disorders medicine*, Springer, 977–996.

Girosi, F., M. Jones, and T. Poggio (1995): "Regularization Theory and Neural Networks Architectures," *Neural Computation*, 7, 219–269.

Goldberger, A. L., L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley (2000): "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals." *Circulation*, 101, E215–20.

Hamida, S. T.-B. and B. Ahmed (2013): "Computer based sleep staging: Challenges for the future," in *2013 7th IEEE GCC Conference and Exhibition (GCC)*, IEEE, 280–285.

Hassan, A. R. and M. I. H. Bhuiyan (2017): "Automated identification of sleep states from EEG signals by means of ensemble empirical mode decomposition and random under sampling boosting," *Computer Methods and Programs in Biomedicine*, 140, 201–210.

He, K., X. Zhang, S. Ren, and J. Sun (2016): "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *Proceedings of the IEEE International Conference on Computer Vision*, 11, 1026–1034.

Hibbs, A. M., A. Storfer-Isser, C. Rosen, C. E. Ievers-Landis, E. M. Taveras, and S. Redline (2014): "Advanced sleep phase in adolescents born preterm." *Behavioral sleep medicine*, 12, 412–24.

Hochreiter, S. and J. Schmidhuber (1996): "Long Short Term Memory," *Memory*, 1–28.

Hornik, K. (1991): "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, 4, 251–257.

Iber, C., S. Ancoli-Israel, A. Chesson, and S. Quan (2007): *The AASM Manual for the Scoring of Sleep and Associates Events: Rules, Terminology and Technical Specifications*, The American Academy of Sleep Medicine.

Itil, T. M., D. M. Shapiro, M. Fink, and D. Kassebaum (1969): "Digital computer classifications of EEG sleep stages." *Electroencephalography and clinical neurophysiology*, 27, 76–83.

Jackson, C. L., S. Redline, I. Kawachi, and F. B. Hu (2013): "Association between sleep duration and diabetes in black and white adults," *Diabetes Care*, 36, 3557–3565.

Kemp, B., A. H. Zwinderman, B. Tuk, H. A. C. Kamphuisen, and J. J. L. Oberyé (2000): "Analysis of a sleep-dependent neuronal feedback loop: The slow-wave microcontinuity of the EEG," *IEEE Transactions on Biomedical Engineering*, 47, 1185–1194.

Kingma, D. P. and J. L. Ba (2015): "Adam: a Method for Stochastic Optimization," *International Conference on Learning Representations 2015*, 1–15.

Längkvist, M., L. Karlsson, and A. Loutfi (2012): "Sleep Stage Classification Using Unsupervised Feature Learning," *Advances in Artificial Neural Systems*, 2012, 1–9.

LeCun, Y., Y. Bengio, and G. Hinton (2015): "Deep learning," *Nature*, 521, 436–444.

Lee-Chiong, T. (2005): *Sleep: A Comprehensive Handbook*, John Wiley and Sons.

Liang, S. F., C. E. Kuo, F. Z. Shaw, Y. H. Chen, C. H. Hsu, and J. Y. Chen (2016): "Combination of expert knowledge and a genetic fuzzy inference system for automatic sleep staging," *IEEE Transactions on Biomedical Engineering*, 63, 2108–2118.

Lipton, Z. C., J. Berkowitz, and C. Elkan (2015): "A Critical Review of Recurrent Neural Networks for Sequence Learning," *arXiv*, 1–38.

Magnin, M., M. Rey, H. Bastuji, P. Guillemant, F. Mauguiere, and L. Garcia-Larrea (2010): "Thalamic deactivation at sleep onset precedes that of the cerebral cortex in humans," *Proceedings of the National Academy of Sciences*, 107, 3829–3833.

Miller, M. and F. Cappuccio (2007): "Inflammation, Sleep, Obesity and Cardiovascular Disease." *Current Vascular Pharmacology*, 5, 93–102.

Nir, Y., R. J. Staba, T. Andrillon, V. V. Vyazovskiy, C. Cirelli, I. Fried, and G. Tononi (2011): "Regional Slow Waves and Spindles in Human Sleep," *Neuron*, 70, 153–169.

Onton, J. A., D. Y. Kang, and T. P. Coleman (2016): "Visualization of Whole-Night Sleep EEG From 2-Channel Mobile Recording Device Reveals Distinct Deep Sleep Stages with Differential Electrodermal Activity," *Frontiers in Human Neuroscience*, 10, 605.

Oostenveld, R., P. Fries, E. Maris, and J.-M. Schoffelen (2011): "FieldTrip: Open Source Software for Advanced Analysis of MEG, EEG, and Invasive Electrophysiological Data," *Computational Intelligence and Neuroscience*, 2011, 1–9.

Penzel, T., K. Kesper, V. Gross, H. Becker, and C. Vogelmeier (2003): "Problems in automatic sleep scoring applied to sleep apnea," *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No.03CH37439)*, 1, 358–361.

Rasch, B. and J. Born (2013): "About Sleep's Role in Memory," *Physiological Reviews*, 93, 681–766.

Rechtschaffen, A. and A. Kales (1968): "A manual of standardised terminology, techniques, and scoring system for sleep stages of human subjects." *Los Angelos: UCLA Brain Information Service .*, 2–115.

Rosen, C. L., E. K. Larkin, H. L. Kirchner, J. L. Emancipator, S. F. Bivins, S. A. Surovec, R. J. Martin, and S. Redline (2003): "Prevalence and risk factors for sleep-disordered breathing in 8- to 11-year-old children: Association with race and prematurity," *Journal of Pediatrics*, 142, 383–389.

Rosenberg, R. S. and S. Van Hout (2014): "The American Academy of Sleep Medicine inter-scorer reliability program: Respiratory events," *Journal of Clinical Sleep Medicine*, 10, 447–454.

Sen, B., M. Peker, A. Cavusoglu, and F. V. Celebi (2014): "A comparative study on classification of sleep stage based on EEG signals using feature selection and classification algorithms," *Journal of Medical Systems*, 38.

Shea, K. O. and R. Nash (2015): "An Introduction to Convolutional Neural Networks," *arXiv*, 1–8.

Shen, D., G. Wu, and H.-I. Suk (2017): "Deep Learning in Medical Image Analysis," *Annual Review of Biomedical Engineering*, 2017, 221–248.

Silber, M. H., S. Ancoli-Israel, M. H. Bonnet, S. Chokroverty, M. M. Grigg-Damberger, M. Hirshkowitz, S. Kapen, S. A. Keenan, M. H. Kryger, T. Penzel, M. R. Pressman, and C. Iber (2007): "The visual scoring of sleep in adults," *Journal of Clinical Sleep Medicine*, 3, 121–131.

Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. V. D. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, and K. Kavukcuoglu (2016): "Mastering the game of Go with deep neural networks and tree search," *Nature*, 529, 484–489.

Simor, P., F. Gombos, S. Szakadát, P. Sándor, and R. Bódizs (2016): "EEG spectral power in phasic and tonic REM sleep: different patterns in young adults and children." *Journal of sleep research*, 25, 269–77.

Spilsbury, J. C., A. Storfer-Isser, D. Drotar, C. L. Rosen, H. L. Kirchner, and S. Redline (2005): "Effects of the home environment on school-aged children's sleep." *Sleep*, 28, 1419–27.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014): "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, 15, 1929–1958.

Supratak, A., H. Dong, C. Wu, and Y. Guo (2017): "DeepSleepNet: a Model for Automatic Sleep Stage Scoring based on Raw Single-Channel EEG," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 4320, 1–1.

Tsinalis, O., P. M. Matthews, and Y. Guo (2016a): "Automatic Sleep Stage Scoring Using Time-Frequency Analysis and Stacked Sparse Autoencoders," *Annals of Biomedical Engineering*, 44, 1587–1597.

Tsinalis, O., P. M. Matthews, Y. Guo, and S. Zafeiriou (2016b): "Automatic Sleep Stage Scoring with Single-Channel EEG Using Convolutional Neural Networks," *arXiv*, 12.

van den Oord, A., S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves,

N. Kalchbrenner, A. Senior, and K. Kavukcuoglu (2016): “WaveNet: A Generative Model for Raw Audio,” *arXiv*, 1–15.

Wang, J. Y., F. D. Weber, K. Zinke, M. Inostroza, and J. Born (2017): “More Effective Consolidation of Episodic Long-Term Memory in Children Than Adults-Unrelated to Sleep,” *Child Development*, 00, 1–15.

Whitney, C. W., D. J. Gottlieb, S. Redline, R. G. Norman, R. R. Dodge, E. Shahar, S. Surovec, and F. J. Nieto (1998): “Reliability of scoring respiratory disturbance indices and sleep staging.” *Sleep*, 21, 749–57.

Wu, Y., M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean (2016): “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation,” *arXiv*, 1–23.

Zhang, J. and Y. Wu (2017): “Automatic sleep stage classification of single-channel EEG by using complex-valued convolutional neural network,” *Biomedical Engineering / Biomedizinische Technik*, 0.

Zhang, J., Y. Wu, J. Bai, and F. Chen (2016): “Automatic sleep stage classification based on sparse deep belief net and combination of multiple classifiers,” *Transactions of the Institute of Measurement and Control*, 38, 435–451.

Zhang, Y., M. Pezeshki, P. Brakel, S. Zhang, C. L. Y. Bengio, and A. Courville (2017): “Towards End-to-End Speech Recognition with Deep Convolutional Neural Networks,” *arXiv*, 410–414.

Appendix

A Network Architecture

Table 7: CNN architecture for use with one channel.

Layer name	Filter-Size	Stride	Output dims	L2	Dropout	BatchNorm
Input			(2800,1)			
Conv 1D	128	5	(551,128)	0.05	0.2	Yes
Conv 1D		1	(547,256)	0.01	0.2	Yes
MaxPool	256	2	(273,256)			
Conv 1D		2	(135,300)	0.01	0.2	Yes
MaxPool	300	2	(67,300)			
FC			(1500,)		0.5	Yes
FC			(1500,)		0.5	Yes
Softmax			5			

Table 8: LSTM architecture with input features from the CNN

Layer name	Output dims	Dropout	Rec. Dropout
Input	(1500,1)		
LSTM	(100,)	0.3	0.3
LSTM	(100,)	0.3	0.3
Softmax	(5,)		

Table 9: feat-ANN architecture

Layer name	Output dims	Dropout	BatchNorm
Input	(37,1)		
FC	(80,)	0.35	Yes
FC	(80,)	0.35	Yes
Softmax	(5,)		

Table 10: feat-LSTM architecture

Layer name	Output dims	Dropout	Rec. Dropout
Input	(37,1)		
LSTM	(80,)	0.3	0.3
LSTM	(80,)	0.3	0.3
Softmax	(5,)		

Declaration of Authorship

I hereby confirm that I have authored this Master's thesis independently and without use of others than the indicated sources. All passages which are literally or in general matter taken out of publications or other sources are marked as such.

Simon Kern
Nijmegen, August 18, 2017