Biomedical Signal Processing and Control 81 (2023) 104496

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

Automatic sleep scoring using patient-specific ensemble models and knowledge distillation for ear-EEG data

Kenneth Borup [a,*], Preben Kidmose [b], Huy Phan [c], Kaare Mikkelsen [b]

[a] Department of Mathematics, Aarhus University, Denmark
[b] Department of Electrical and Computer Engineering, Aarhus University, Denmark
[c] School of Electronic Engineering and Computer Science, Queen Mary University of London, United Kingdom

ARTICLE INFO

Keywords:
Automatic sleep scoring
Ensemble models
Knowledge distillation
Light-weight sleep scoring
Semi-supervised learning
Personalized models

ABSTRACT

Human sleep can be described as a series of transitions between distinct states. This makes automatic sleep analysis (*scoring*) suitable for an automatic implementation using machine learning. However, the task becomes harder when data is sampled using more light-weight or mobile equipment, often chosen due to greater comfort for the patient. In this study we investigate the improvement in sleep scoring when multiple state-of-the-art neural networks are joined into an ensemble, and subsequently *distilled* into a single model of identical network architecture, but with improved predictive performance. In this study we investigate ensembles of up to 10 networks, and show that, on the same data, ensembles of neural networks perform better than each single subject model (improvement: 2.4%) and that this improvement can be transferred back into a single network using a combination of patient specific data and *knowledge distillation*.

The study demonstrates both a way to further improve automatic sleep scoring from mobile devices, which in itself is interesting, but also highlights the great potential of the vast amounts of unlabeled personal data which will become available from personal recording devices.

1. Introduction

Humans sleep for one third of their lives. Our sleep both affects and is impacted by our health, and as such knowledge about patient sleep is recognized as a valuable ingredient in clinical care and diagnosis [1,2]. However, the current gold standard for sleep monitoring builds on manual classification (*scoring*) of *polysomnography* (PSG) recordings, the entire process of which is both expensive and intrusive on the patient's actual sleep. This has lead to repeated attempts to update the process, both through new and more light-weight recording devices [3–7] and automatic algorithms for analyzing the data [8,9]. The present study falls in both categories, in that we explore automatic algorithms specifically for scoring light-weight recordings.

Multiple studies [4,10,11] have shown the efficiency of ensemble models for classification of electroencephalography (EEG) data. At the same time, all state-of-the-art algorithms for automatic sleep scoring in the past few years have been built on neural networks [12]. A natural question then becomes to which extent the combination of these methods will lead to even better performance?

In this paper we train ensembles of neural networks for sleep scoring, and perform a thorough investigation of the possible benefits and realistic applications of this method. We focus on an established deep neural network for automatic sleep scoring, the SeqSleepNet [8], in the specific context of a proven light-weight sleep monitoring technology, the ear-EEG [13]. A major limitation of neural ensembles is the added memory and computational requirements for such models. This leads us to a further investigation of the benefits of *knowledge distillation* [14–16], which was specifically introduced to alleviate this problem by *distilling* the ensemble of models into a single model at the cost of a small loss in predictive performance.

The idea of *knowledge distillation* (or just *distillation*) originates back to Bucila et al. [16], and was later brought to the deep learning setting by Ba and Caruana [15], but it is most commonly known as a model compression technique popularized by Hinton et al. [14]. It is a procedure to transfer some statistic (often called *knowledge*) from one model (teacher) to another model (student). Originally the student was considered smaller[1] than the teacher, and the distillation procedure aimed at training the student to mimic the softened probability

* Corresponding author.
E-mail addresses: kennethborup@math.au.dk (K. Borup), pki@ece.au.dk (P. Kidmose), h.phan@qmul.ac.uk (H. Phan), mikkelsen.kaare@ece.au.dk (K. Mikkelsen).

[1] Depending on the task at hand different measures of model size can be relevant; e.g. model parameters, inference time, memory requirements or model complexity. However, often model parameters is considered a reasonable proxy for model size.

https://doi.org/10.1016/j.bspc.2022.104496
Received 11 May 2022; Received in revised form 3 October 2022; Accepted 1 December 2022
Available online 16 December 2022

distribution over the logits of the (trained and fixed) teacher model alongside the original training data. However, since the formulation by Hinton et al. [14] an extensive amount of alterations to the procedure has been proposed. A branch of research propose mimicking the teacher on other statistics than the distribution of logits [17–21], while another branch focus on developing the transfer procedure and the choice of data used for distillation [22–29].

Exactly why knowledge distillation works well is still an open research question, and an active field of research, but Mobahi et al. [30] show that self-distillation[2] with kernel ridge regression models progressively shrinks the number of basis functions used to represent the solution, thus acting as a method of regularization. Furthermore, Borup and Andersen [31] show that this behavior is highly dependent on the weighting between labeled ground-truth data and teacher outputs used during distillation. Our application of knowledge distillation build on the empirical successes of distillation techniques and is closely related to self-distillation.

In order to reduce the computational burden of ensemble models at inference time, we utilize knowledge distillation to distill the cumbersome ensemble into a single model. Our distillation framework is very flexible, and distillation can be performed in supervised, semi-supervised or unsupervised settings depending on the available data.

We find that forming ensembles of neural networks does indeed improve performance relative to single networks, and that by using unlabeled data from the individual patient, we can transfer some of that improvement back into a single network using knowledge distillation.

Our contributions In this study we present a number of important contributions to the field of automatic sleep scoring:

- To our knowledge, this is the first study successfully leveraging unlabeled, personal data, which is likely to be important in long term sleep monitoring.
- We show that simple ensembles of 10 SeqSleepNet models trained independently improve predictive performance, and only differ by 0.04 in Cohen's kappa compared to the best case scenario of two manual scorers.
- Despite no change in model architecture, we show that a single SeqSleepNet model trained with our semi-supervised distillation setup retains between 50% and 100% of the improvements obtained by ensemble models (of various size) when trained with personal data.

Details on our experimental setup can be found in Supplementary Material A, and code to reproduce our experimental results is publicly available at github.com/Kennethborup/SeqSleepNet.

2. Problem setup and methods

2.1. Data

In this study, the input data to the algorithm is a bilateral ear-EEG derivation (specifically, the average of the left ear electrodes relative to the average of the right ear electrodes), while labels come from a manual scoring of a reduced PSG montage. See Fig. 1 for visualizations of the two methods. The *left–right* ear-EEG derivation is used because it has been thoroughly studied for sleep scoring, and has been shown to be a strong candidate for clinical-grade home sleep monitoring [32]. We recommend reading Mikkelsen et al. [4] for a detailed description of the recording platform.

The specific recordings used are presented in Mikkelsen et al. [4] and Mikkelsen et al. [33]. Combined, they constitute a data set of 20 subjects recorded using the same equipment. Each subject has four nights of labeled recordings. Half of the subjects also have a further 12 nights of unlabeled recordings each. We shall refer to the two groups of subjects as respectively *short* and *long* subjects. See Fig. 2 for an overview.

In accordance with standard sleep scoring practice, the sleep recordings have been partitioned into 30-second epochs. For the labeled recordings, they have been manually scored by the same sleep technician according to the five-stage scoring described in the AASM manual [1]: *Wake*, *REM*, *Non-REM 1*, *Non-REM 2*, and *Non-REM 3*.

2.2. Cohen's kappa score

As is established practice when quantifying sleep scoring performance, we measure the performance of our automatic classifier by calculating *Cohen's kappa* [34] between the predicted and manual labels.

2.3. Model architecture (SeqSleepNet classifier)

In this paper we use the sequence-to-sequence neural network architecture SeqSleepNetintroduced in Phan et al. [8]. SeqSleepNet takes a sequence of L consecutive epochs as input and outputs a sequence of L 5-dimensional probability vectors. The input can be either single- or multichannel log-scale spectrograms, where the data of each channel is (approximately) normalized to zero mean and unit variance for each frequency bin. The output probability vectors are the predicted class-probability for each of the $P = 5$ sleep stages. In this paper we follow the settings of Mikkelsen et al. [32] and use spectrograms with $T = 29$ time bins (spanning 30 s), with $F = 129$ frequency bins, and with a single, $C = 1$, channel. Furthermore, we will use a sequence length of $L = 20$ as in Mikkelsen et al. [32] and Phan et al. [8]. We denote each epoch by $\mathbf{z}_i \in \mathbb{R}^{T\times F\times C}$, and the sequence of L epochs by $\mathbf{x}_n = (\mathbf{z}_n, \mathbf{z}_{n+1}, \dots, \mathbf{z}_{n+L-1}) \in \mathbb{R}^{L\times T\times F\times C}$.[3] For more details on the SeqSleepNet architecture we refer to Phan et al. [8] and our implementation in PyTorch available at GitHub: github.com/Kennethborup/SeqSleepNet. For details on training we refer to Section 2.4 and to Appendix for experimental details and additional results.

Sliding window average prediction. The sequence-to-sequence nature of the SeqSleepNet allows us to obtain predictions on a sequence of epochs with a sliding window approach. More specifically, we first apply our model on $\mathbf{x}_1 = (\mathbf{z}_1, \dots, \mathbf{z}_{1+L-1})$ followed by $\mathbf{x}_2 = (\mathbf{z}_2, \dots, \mathbf{z}_{2+L-1})$ and so on. Thus, by *sliding* our model across a sequence of L epochs by increments in index of one, we obtain L predictions for each epoch, and averaging these predictions for each epoch yields a new probability vector.[4] Throughout this study, we will always be utilizing this sliding window average and therefore refer to this procedure merely as predicting. Note, that while this procedure improves predictive performance, it also requires L times as many steps of predictions, which is computationally expensive at inference time, especially for large L.

[2] Self-distillation often refers to the use of identical teacher and student models, which is not entirely true for our setup as our teacher is an ensemble.

[3] When using multiple nights for training or evaluation, we assume no gap between the nights, and thus, some sequences of epochs might overlap two different nights or even subjects. However, the effect of this on the overall predictive performance is low from our experience.

[4] Note, for the initial and final $L-1$ epochs we will not obtain L predictions, due to missing data prior and after the sequence, and we will merely consider the average predictions of all the possible predictions at these epochs.

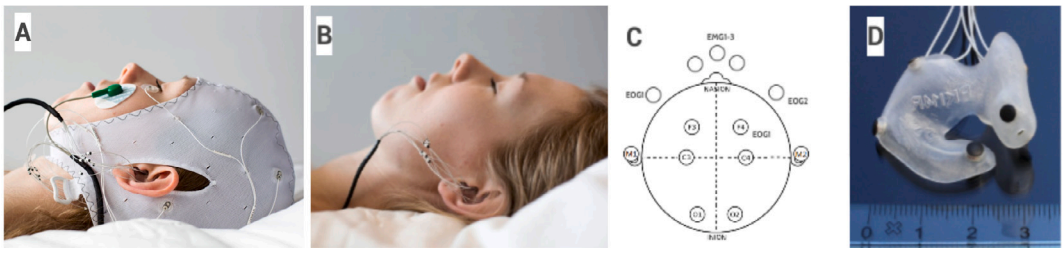


Fig. 1. The recording setup used in the data set. **(a)**: the setup used for the labeled recordings, where the data from the electrodes in the cap is used for the manual labeling. **(b)**: the setup used for the unlabeled recordings. There are only electrodes inside the ears and next to the right eye. **(c)**: Positions of the electrodes in picture **(a)**, excluding the ear electrodes. Note the two positions next to the eyes and the three on the chin. **(d)**: an example of the soft ear pieces with dry electrodes placed inside the ears. Note that the ruler at the bottom goes from 0 to 3 cm.

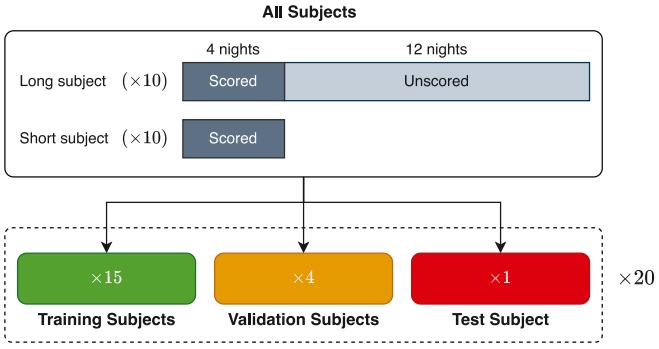


Fig. 2. We perform our experiments using Leave-One-Subject-Out-Cross-Validation (LOSO-CV), and our dataset consists of 20 subjects. 10 subjects (denoted *short* subjects) has 4 nights of scored observations, and the remaining 10 subjects (denoted *long* subjects) has 4 nights of scored observations along with 12 nights of *unscored* observations. For each CV-step we divide the subjects into a training (15 subjects), validation (4 subjects) and test (one subject) set, irrespective of the subject type (long/short). Thus, each set can consist of both long and short subjects, but whether the unscored recordings are used or not, depend on the particular experiment.

2.4. Model training

We perform our experiments using Leave-One-Subject-Out-Cross-Validation (LOSO-CV) over all 20 subjects (both *short* and *long*). For each CV-step we divide the subjects into training (15 subjects), validation (4 subjects) and test (one subject) sets, irrespective of the subject-type (long/short) - see Fig. 2 for an illustration of this. Thus, for each random initialization of our model we train 20 different models, but will merely refer to it as one model and will report the predictive performance of this model as the average Cohen's kappa on the 4 scored nights of the test subjects across all 20 subjects. Our study is split in two phases; in Phase 1, we train a set of single SeqSleepNet models in the classical supervised way, and denote these models as *baseline* models. In Phase 2, we collect these baseline models into a large and computationally demanding ensemble model (called a *teacher* model) which in turn is distilled into a single SeqSleepNet model by utilization of knowledge distillation, thereby reducing the computational requirements at inference time significantly.

Baseline models and training (Phase 1) We independently train M random initializations of the SeqSleepNet model, and refer to these models as baseline models, each denoted by $\mathcal{B}_j$ for $j = 1, \dots, M$. We train each model to minimize the cross-entropy loss on the 4 scored nights for all training subjects. That is, let $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N}$ be the training dataset, then we minimize

$$\mathcal{L}_{\text{CE}}(\mathcal{B}_j) \stackrel{\text{def}}{=} \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}_{\text{train}}} \ell_{\text{CE}}(\mathbf{y}, \mathcal{B}_j(\mathbf{x})), \quad \text{where}$$

$$\ell_{\text{CE}}(\mathbf{t}, \mathbf{s}) \stackrel{\text{def}}{=} -\sum_{l=1}^{L}\sum_{p=1}^{P} [\mathbf{t}_l]_p \log\left([\mathbf{s}_l]_p\right), \quad \text{for} \quad \mathbf{t}, \mathbf{s} \in \mathbb{R}^{L\times P},$$

and where $\mathcal{B}_j(\mathbf{x})$ is the sequence of predicted class-probabilities on $\mathbf{x}$, and $\mathbf{y}$ the sequence of associated one-hot encoded ground-truth labels. We refer to training of the baseline models as supervised training or Phase 1 (see Fig. 3) and remind that by $\mathcal{B}_j$ we in fact refer to the 20 underlying models trained in a LOSO-CV setup.

Ensemble models Based on the M baseline models, $\{\mathcal{B}_j\}_{j=1}^{M}$, we can construct ensemble models of size m, where $m \in \{1, \dots, M\}$ is the amount of baseline models used in the ensemble. We construct the ensemble models as the unweighted average of the m individual predictions on some sample $\mathbf{x}$, i.e. as $\mathcal{T}(\mathbf{x}) = \frac{1}{m}\sum_{j=1}^{m}\mathcal{B}_j(\mathbf{x})$, and thus $\mathcal{T}(\mathbf{x})$ is still a probability vector. We denote an ensemble model of size m by $\mathcal{T}_m$, or merely $\mathcal{T}$ if the size is unambiguous.[5] By using an unweighted average ensemble of baseline models, no additional training is required to construct the ensemble, but m times more prediction steps are required in order to perform inference. Due to the L times more steps required by the sliding window prediction of each baseline model, prediction with an ensemble model requires Lm times more prediction steps compared to naïve prediction using a single baseline model. In Section 3 we report the predictive performance of all possible ensemble models constructed of unique sets of m baseline models, and in the following we investigate a semi-supervised adaptation of knowledge distillation as a way to reduce the computational requirements of ensemble models at inference time. We stress the fact, that distilling an ensemble of m baseline models into a single student model, reduces the computational requirements at inference time by $m\times$ at a small loss in predictive performance (see e.g. Fig. 4).

2.5. Distillation of ensembles to single models (Phase 2)

In the following we present our approach to distillation which allow for utilization of unlabeled data in a semi-supervised manner. This approach is at large similar to methods sometimes known as self-training or self-distillation.

In this study we utilize a semi-supervised adaptation of the original knowledge distillation technique, where we match the teacher on a set of unlabeled data, and employ an imbalanced smoothing of the labels — see below for details. Thus, we now refer to $\mathcal{T}$ as the *teacher* model, and initialize a new SeqSleepNet model denoted by $\mathcal{S}$ which we refer to as the *student* model following the conventions in the knowledge distillation literature. Let $\mathcal{D}_{\text{distill}} = \mathcal{D}_{\text{gt}} \cup \mathcal{D}_{\text{pseudo}}$ be the *distillation dataset*, where $\mathcal{D}_{\text{gt}} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N_{\text{gt}}}$ and $\mathcal{D}_{\text{pseudo}} = \{(\mathbf{x}_n, \mathcal{T}(\mathbf{x}_n))\}_{n=1}^{N_{\text{pseudo}}}$ are the ground-truth and pseudo-labeled data sets, respectively. We will refer to the predictions of the teacher, $\mathcal{T}(\mathbf{x})$, on the pseudo-labeled dataset, $\mathcal{D}_{\text{pseudo}}$, as *pseudo-labels*. Note, the set of input samples for $\mathcal{D}_{\text{gt}}$ and $\mathcal{D}_{\text{pseudo}}$ need not be equal, and are often disjoint. Furthermore,

[5] We remind that by $\mathcal{T}_m$ we in fact refer to the 20 underlying models trained in a LOSO-CV setup. Thus, each of the 20 underlying ensemble models are the unweighted combination of the m underlying baseline models at the particular CV-step.

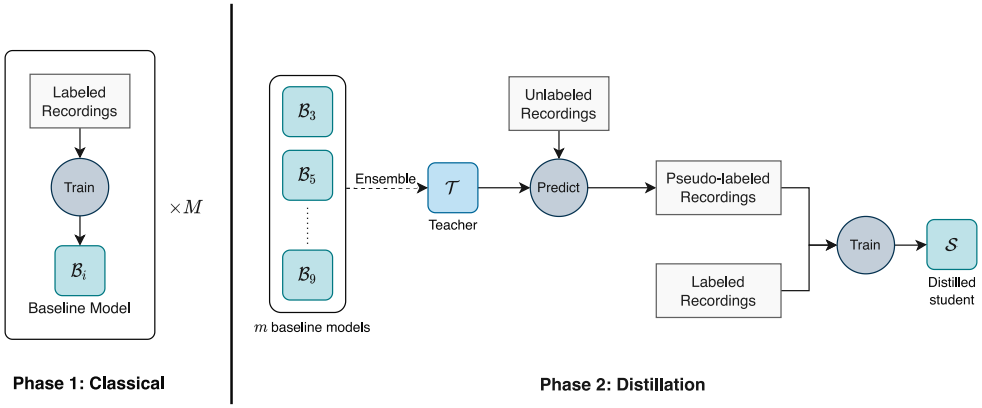


Fig. 3. Our training procedure is split into two phases: (1) classical training of baseline models and (2) distillation of ensembles of baseline models into a single student model. In Phase 1 we independently train M random initializations of the baseline model on the 15 training subjects (note, the set of training subjects depend on the initialization, but the test subject is constant.). We denote each of these trained models by $\mathcal{B}_j$ for $j = 1, \ldots, M$. In Phase 2, we combine a subset (of size m) of baseline models into an ensemble model which we use as a teacher (denoted by $\mathcal{T}$). We obtain pseudo labels on a selection of unscored data (which can be from training, validation and/or test subject(s)) as predictions from $\mathcal{T}$ and train the student model (denoted by $\mathcal{S}$) on these pseudo labels as well as optionally (hard) labeled training data.

$\mathcal{D}_{\text{pseudo}}$ does not require any labels allowing for a semi- or unsupervised distillation procedure. Define the distillation-loss as a weighted (by $\alpha \in [0, 1)$) sum of two terms; one for scored samples and one for pseudo-labeled samples, i.e. as

$$\mathcal{L}_{\text{distill}}(S) \overset{\text{def}}{=} \alpha \mathcal{L}_{\text{gt}}(S) + (1 - \alpha)\mathcal{L}_{\text{pseudo}}(S) \tag{1}$$

where $\mathcal{L}_{\text{gt}}$ and $\mathcal{L}_{\text{pseudo}}$ are the ground-truth and pseudo loss respectively, and are defined as

$$\mathcal{L}_{\text{gt}}(S) \overset{\text{def}}{=} \frac{1}{|\mathcal{D}_{\text{gt}}|} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_{\text{gt}}} \ell_{\text{CE}}\left(\mathbf{y}, S(\mathbf{x})\right), \quad \text{and} \tag{2}$$

$$\mathcal{L}_{\text{pseudo}}(S) \overset{\text{def}}{=} \frac{1}{|\mathcal{D}_{\text{pseudo}}|} \sum_{(\mathbf{x},\mathcal{T}(\mathbf{x})) \in \mathcal{D}_{\text{pseudo}}} \ell_{\text{CE}}\left(\boldsymbol{\sigma}\left(\tilde{\mathcal{T}}(\mathbf{x})/\tau\right), S(\mathbf{x})\right), \tag{3}$$

where σ is the softmax function, $\tilde{\mathcal{T}}(\mathbf{x})$ is the pre-softmax logits of $\mathcal{T}(\mathbf{x})$ (i.e. $\sigma(\tilde{\mathcal{T}}(\mathbf{x})) = \mathcal{T}(\mathbf{x})$), $\alpha \in [0, 1)$ is a weighting parameter, and τ a temperature for softening/sharpening of the teacher class-probabilities introduced in Hinton et al. [14].[6] Setting $\alpha = 0$ (and ensuring $\mathcal{D}_{\text{pseudo}} \neq \emptyset$) makes the distillation procedure fully unsupervised, while $\alpha \in (0, 1)$ yields a semi-supervised procedure. In this paper we consistently use $\alpha = 0.5$ and $\tau = 1$.[7] Hence, the distillation procedure is as follows: (1) fix the teacher, $\mathcal{T}$, (2) compute pseudo-labels with $\mathcal{T}$, and (3) train the student, S, on the distillation dataset, $\mathcal{D}_{\text{distill}}$, by minimizing $\mathcal{L}_{\text{distill}}(S)$ in (1). See Phase 2 in Fig. 3 for an illustration of the distillation procedure. Note, when distilling a teacher, $\mathcal{T}_m$, to a single baseline model we reduce the computational requirements at inference by m times which for e.g. $m = 10$ corresponds to a decrease of 90%.

Since no labels are used for $\mathcal{D}_{\text{pseudo}}$, we can use both scored (i.e. discarding the known labels) and truly unscored samples in $\mathcal{D}_{\text{pseudo}}$ as well as samples from validation and/or test subjects. When any data from the test subject is used during distillation, we refer to the student as a *personalized* student, and a *general* student otherwise.

3. Results

We summarize selected results and baseline results in Table 1, and have collected confusion matrices in Fig. 6. We find from the confusion matrices that the performance improvements, when going from worst performing model (baseline) to best performing (10-model ensemble), is spread across all 5 stages rather than a specific stage getting better.

Below, we have separated the analysis of our results into separate segments for specific model groups: Ensemble teachers, general students and personalized students.

3.1. Ensemble teachers

We let $M = 10$, and in Fig. 4 we report the mean test performance of all simple ensemble models constructed of unique sets of m baseline models for $m = 1, \ldots, 10$ along with the empirical 25–75% and 10–90% confidence interval for each m.[8] In total we consider 1023 different ensemble models. We see a monotonic improvement in mean predictive performance with increasing m, where the performance increase is largest for small m, and the performance appears to saturate at $\approx$ 0.780. Furthermore, there exists ensembles with $m \geq 4$ that perform equivalently to the best performing ensemble with $m = 10$ (selecting these specific ensemble models prior to training and evaluation of all ensemble models is not possible). Compared to our baseline at 0.755, which is equivalent to previous state-of-the-art on this dataset [32], an ensemble of merely two models improves the mean performance by 0.013, while an ensemble of 10 models improves by 0.024.

3.2. Distilled students

In the following we let $\mathcal{D}_{\text{gt}}$ be the set of all training subjects and investigate the performance of student models trained with the distillation procedure from Section 2.5 for different choices of $\mathcal{D}_{\text{pseudo}}$. More specifically, we separately consider the case where *no* data from the test subject is used (*general students)*, and the case where some data from the test subjects is used to personalize the student (*personalized students*). We investigate the impact on the student performance by the size (i.e. m) of the teacher model as well as the particular distillation dataset chosen. We repeat all experiments four times with different seeds, and report the mean performance across all four replications. See Appendix for experimental details and used hyperparameters.

[6] Note, unlike classical distillation, we do not apply the temperature softening to the student logits, but only the teacher logits creating an *imbalanced* setting.

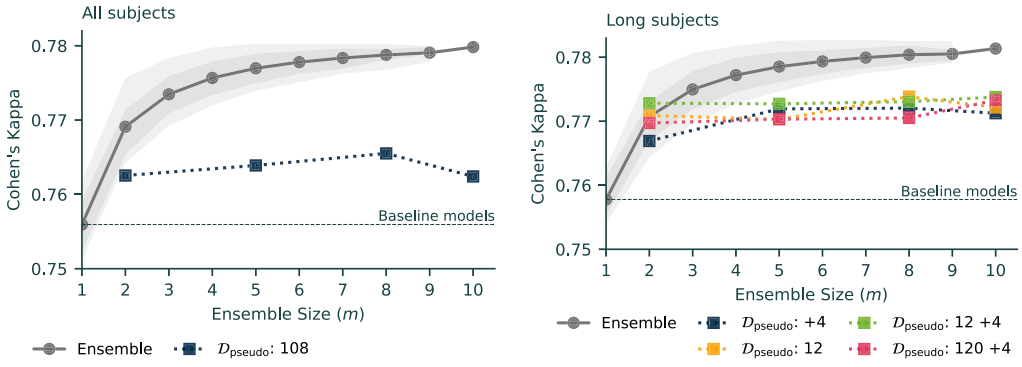
[7] By investigation of different choices of τ, we find that the performance does not change much for τ between 0.1 and 1, and $\tau \leq 2$ yield some improvement — see Fig. 8. Thus, we consistently use $\tau = 1$. Furthermore, we leave the investigation of varying α to future work.

[8] Note, the amount of possible models vary with m. I.e. with 10 baseline models, then for $m = 1$ there are 10 possible ensemble models, for $m = 2$ there are 45 models, for $m = 3$ there are 120 models and so on.

Table 1
Here General Student is trained merely on data from other subjects (see Section 3.2.1), while Personal Student is trained on the additional 12 unlabeled subject-specific nights (see Section 3.2.2). We also report a subset of Ensemble models, the Baseline introduced in this paper, as well as the state-of-the-art on the labeled data alone. The standard deviation is reported in parentheses (estimated across 4 replicated experiments for the distilled students and across all possible ensembles).

	Models	Teacher size (m)	Personal data	Unlabeled data	Inference time[a] (S)	Long subjects	Short subjects	All subjects
Baseline (ours)	1	–	–	–	10.3 (1×)	0.758 (±0.005)	0.754 (±0.011)	0.756 (±0.006)
Ensemble	2	–	–	–	20.6 (2×)	0.771 (±0.005)	0.767 (±0.006)	0.769 (±0.004)
Ensemble	10	–	–	–	102.8 (10×)	0.781 (±0.000)	0.778 (±0.000)	0.780 (±0.000)
General Student	1	2	–	✓	10.3 (1×)	0.766 (±0.008)	0.759 (±0.005)	0.763 (±0.006)
General Student	1	8	–	✓	10.3 (1×)	0.769 (±0.003)	0.762 (±0.003)	0.766 (±0.002)
Personal Student	1	2	✓	✓	10.3 (1×)	0.771 (±0.005)	0.753 (±0.007)	0.762 (±0.005)
Personal Student	1	8	✓	✓	10.3 (1×)	0.774 (±0.002)	0.749 (±0.008)	0.761 (±0.004)
Mikkelsen et al. [4]	1	–	–	–	–	–	–	0.73[b]
Mikkelsen et al. [4]	1	–	✓	–	–	–	–	0.76[b]
Mikkelsen et al. [32]	1	–	–	–	–	–	–	0.76

[a]Inference time is measured as the average time over 100 samples on an Apple M1 Pro CPU and extrapolated to a night of 8 h of sleep recordings.
[b]Mikkelsen et al. [4] report median performance rather than mean, and due to the left tail of the distribution, the median is larger than the mean for these reported results.



(a) All 20 subjects with general students. (b) Only 10 long subjects with personalised students.

Fig. 4. Cohen's kappa for general **(a)** and personalized **(b)** student models. The x-axis is the number of baseline models used in the teacher ensemble, but all students are merely constructed as a *single* model, and the position of students on the x-axis indicates the amount of models in the teacher model. The light and dark shaded gray areas represent the 10–90% and 25–75% empirical confidence intervals.

3.2.1. General students

We now consider the case where $\mathcal{D}_{\text{pseudo}}$ is the set of 108 unscored training and validation nights.[9] In Fig. 4(a) we report the mean performance for teachers of size $m = 2, 5, 8$, and 10.

We are able to recover about 40% of the improvement obtained by the best teacher in a single student model using our distillation procedure and additional data, which yields an improvement of approx. 0.01 in predictive performance compared to the baseline. In order to verify that our distillation procedure is in fact useful, we compute a weight-space ensemble of the 10 baseline models; that is, for each layer we average the weights of the layer across all baseline models and use these averaged weights in a single SeqSlepNet model. Similar approaches to weight-space ensembles have shown great potential by Izmailov et al. [35],Garipov et al. [36]. However, the weight-space ensemble only perform on par with the single baseline models with Cohen's kappa of 0.759 across all 20 subjects.

3.2.2. Personalized students

In the following section we consider the case where a personalized student model is trained based on a set of unscored observations from the test subject. Thus, we only include the predictive performance on the 10 long subjects in this section. For evaluation of the models on the short subjects, we refer the reader to Fig. 10 in the appendix. Note, at no point do we use any manual scorings from the test subject. We consider the cases where we have access to either the 12 unscored nights, the 4 scored nights (without manual scores) or all 16 nights for the long test subject. In Fig. 4(a) we report the mean performance for teacher ensembles of size $m = 2, 5, 8$, and 10. If we use an ensemble of size $m = 2$ personalized students perform equivalently to the teacher at a reduction of 50% in computational costs. Thus, using the distillation procedure we are able to get personalized students that improve by ≈ 0.01 in Cohen's kappa. However, larger teacher ensembles yield only small improvements in personalized student performance. The choice of subject-specific data does not appear to be important, as long as some subject-specific data is used for distillation. This is also supported by Fig. 9 in the Appendix.

In Fig. 5 we compare the performance, on a subject-level, of the baseline models, the ensemble with $m = 10$, and the personalized student based on the 12 unscored nights of the test subject. We sort the subjects by increasing baseline performance, and note that the teacher ensemble consistently outperforms the baseline models on all subjects. Furthermore, the personalized students perform at least as well as the baseline, and even surpass the teacher ensemble for some subjects, despite requiring 1/10'th of the compute at inference time (See Table 1).

More subject-specific data is better In Fig. 4(b) we observe that using the 12 unscored nights from the test subject improves the performance enough to be comparable to the ensemble teacher for some $m > 1$.

[9] For long subjects all 108 unscored training and validation nights are used, while for short subjects (for which there are 120 unscored training and validation nights) the unscored nights of one randomly chosen long subject is not included in $\mathcal{D}_{\text{pseudo}}$, which yield a total of 108 unscored nights for all subjects.

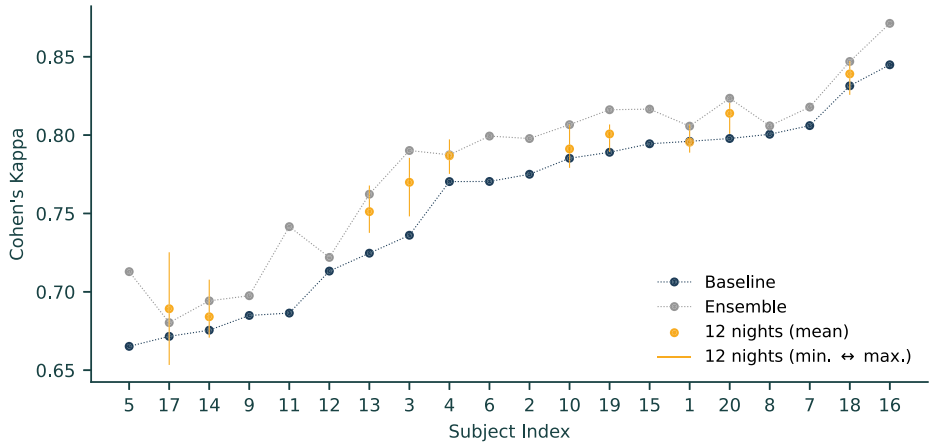
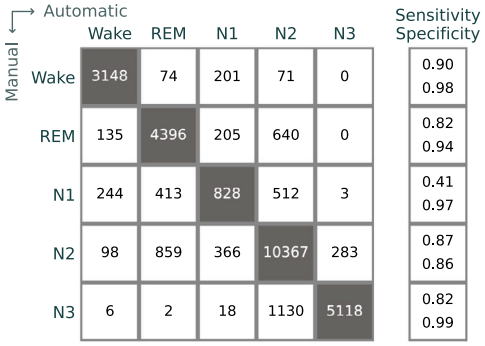
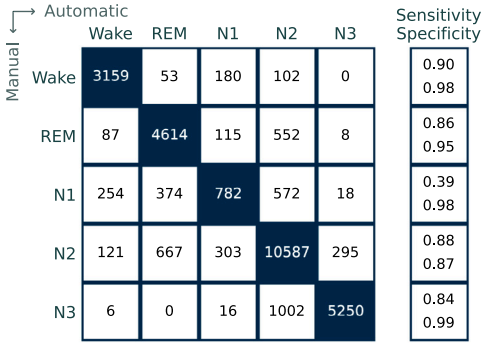


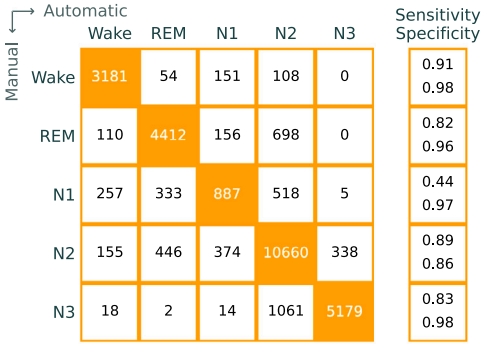
Fig. 5. Cohen's kappa on subject-level, sorted by increasing baseline performance. We report the mean of all 10 baseline models (in blue), the ensemble model of 10 baseline models (in gray), and the mean of the personalized student trained on the 12 unscored nights of the test subject (in yellow with vertical lines between min. and max. of all 8 repetitions). Note, we only report the performance of the personalized student on the long subjects.



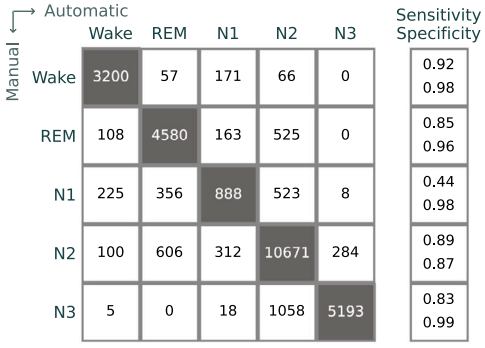
(a) Baseline Model ($m = 1$), $\kappa = 0.763$.



(b) General Student, $\kappa = 0.773$.



(c) Personalized Student, $\kappa = 0.772$.



(d) Ensemble ($m = 10$), $\kappa = 0.781$

Fig. 6. Confusion matrices for comparison between manual scoring (along the y-axis) and automatic scoring (along the x-axis). The values represent the number of epochs and are computed over all long subjects for the five sleep stages. Furthermore, we include the specificity and sensitivity for each class on the right. We report the confusion matrices for (a) a single baseline model trained in a classical supervised manner, (b) a general student trained with pseudo-labels (computed by an ensemble of $m = 8$ models) on the additional 108 unscored nights of non-test subjects, (c) a personalized student trained with pseudo-labels (also computed by an ensemble of $m = 8$ models) on the additional 12 unscored nights associated with the test subject, and (d) the ensemble of $m = 10$ baseline models. We include Cohen's kappa for the long subjects of the models in the caption.

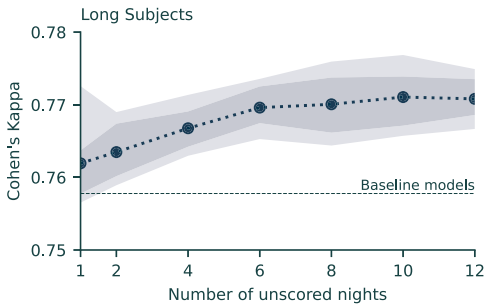


Fig. 7. Cohen's kappa for personalized students trained using the scored training nights along with a varying number of unscored nights from the test subject. We consider merely the performance on the 10 long subjects, and use a teacher with $m = 5$. We report the mean along with 10–90% and 25–75% empirical confidence intervals in shaded areas.

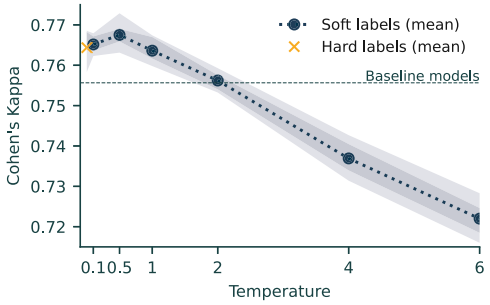


Fig. 8. Different choices of temperature, τ, when student is trained on all unscored data (120 nights) and the 15 scored training subjects. We report the performance over all 20 subjects and the points are the mean of 5 repetitions of the experiments with empirical confidence intervals in shades. Hard labels refer to one-hot encoded predictions by the teacher.

In Fig. 7 we show the personalized student performance when using an increasing number of unscored nights (from one to all 12 nights). We repeat the experiment 10 times with a fixed teacher of size $m = 5$, and observe a near monotonically increasing mean performance with the increase in number of nights. Thus, the more nights available to personalize the model to the test subject, the better.

Importance of adjusting pseudo-labels with a temperature We investigate the effect of changing the temperature, τ, in (3) and plot the performance in Fig. 8. We observe that for $\tau \leq 2$ the performance exceeds the baseline, and more specifically we observe small differences in performance for $\tau \in \{0.1, 0.5, 1\}$ as well as hard pseudo-labels, although with a slight peak at $\tau = 0.5$. Hard pseudo-labels corresponds to letting $\tau \to 0$, and in practice we use the one-hot encoded label with one at the largest entry of the pseudo-label and zero elsewhere. For larger temperatures the performance decreases significantly compared to the baseline.

4. Discussion and conclusion

In this study we have analyzed the utility of ensembles of neural networks (*neural ensembles*) for automatic sleep scoring using wearable EEG recordings. Neural ensembles appear as a likely source of improvement for a difficult problem, making the question suitable for a thorough analysis. On first approach, we find that for the size of data set available here, ensembles are a good approach, and we find a respectable improvement in Cohen's kappa when using ensembles of 10 networks, from 0.756 to 0.780, but also note that with the correct choice of baseline models, even 4–5 models is sufficient for this improvement. Crucially, we find an improvement in all individuals, meaning that, apparently, the ensemble is always better. Given that kappa values between manual scorers are around 0.82 for this data set [32], we think that the improvement shown is close to the best case scenario. We note that our set of baseline models is restricted to be models of identical architecture and training method, but with different random initializations. On this basis, constructing ensembles using baseline models from various different network architectures, of various size, and with stacking are interesting future directions of research in order to improve the predictive performance of the ensembles even further. Furthermore, one can consider reducing the computational requirements at inference time by using cascades of models.

One benefit of mobile sleep monitoring is that the sleep analysis could conceivably be performed locally, for instance on a smartphone. In that case, it is beneficial to keep memory and computation requirements to a minimum, particularly after the model has been trained. Neural ensembles is a potentially very greedy approach which leads us to consider *knowledge distillation*, as a way to compress an ensemble into a single network. We find that the distillation is relatively successful, and a single network can inherit more than half the improvement in kappa value seen for the best teaching ensemble, but using only the same resources as the original baseline model. However, crucially, we find that this degree of improvement requires use of (unlabeled) recordings from the individual for which the model is needed. This is not necessarily a significant issue for mobile sleep monitoring (in which personal unlabeled data is easy to come by), but it is important to keep in mind. We find that the kappa value monotonically increases with the number of recorded nights from the individual, but even without any recordings from the individual, distillation is still able to recover about 40% of the improvement in kappa value, when recordings from other individuals are used.

Moving forward, we find that this is a promising approach for *ambulant sleep monitoring*, and shows an interesting way to benefit from the large amounts of unlabeled data which can easily be gathered in this way. We could imagine a process where the first n nights for a patient were uploaded to a central server in charge of performing the personalized distillation. Once the distillation was completed, the improved, personal model could be returned to the recording device, after which sleep scoring of improved quality could be performed locally. This full process would require no manually scored labels for the patient. It will be interesting in future studies to see whether the personalized benefits could be even larger for more challenging data sets (such as elderly people), where the room for improvement is greater.

CRediT authorship contribution statement

Kenneth Borup: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing , Visualization. **Preben Kidmose:** Conceptualization, Data curation, Supervision. **Huy Phan:** Software. **Kaare Mikkelsen:** Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing, Supervision, Data curation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Kaare Mikkelsen reports financial support was provided by Innovation Fund Denmark. Kaare Mikkelsen reports equipment, drugs, or supplies was provided by T&W Engineering. Preben Kidmose reports equipment, drugs, or supplies was provided by T&W Engineering.

Data availability

Data will be made available on request.

Acknowledgments and disclosure of funding

The work presented here was in part supported by the Innovation Fund Denmark, grant 7050-00007. We thank Lars N. Andersen for comments and suggestions, as well as T&W Engineering for supplied equipment.

Appendix. Experimental details

In the following we present some details on the experimental setting and the hyperparameters used for fitting our models.

A.1. Hyperparameters and training setup

All our models are based on a PyTorch implementation of the SeqSleepNet architecture introduced in Phan et al. [8]. Our code is publicly available at GitHub: github.com/Kennethborup/SeqSleepNet. We set our sequence length to $L = 20$, and consider input epochs of length $T = 29$ with $F = 129$ frequency bins and a single $C = 1$ channel spectogram. More specifically, we only use the *LR* derivation from Mikkelsen et al. [4]. We halve the learning rate when the validation loss has not improved for 50 training epochs and employ early stopping after a minimum of 700 training epochs. Training is done with the Adam optimizer, with learning rate of 10^{-3}, momentum parameters $(\beta_1, \beta_2) = (0.9, 0.999)$, and weight-decay of 10^{-4}. We define a training epoch as the number of gradient updates required to pass through the scored training set (15 subjects each with 4 nights) once, and when training on larger datasets (due to unlabeled data), we still consider a single training epoch to be this amount of steps, but will only pass through a random subset of the samples of the larger dataset. This way, the amount of training epochs are comparable across models, and we limit it to a maximum of 1500 training epochs. However, due to early stopping, training is often effectively stopped at less than 1000 training epochs.

A.2. Ablation of temperature

When performing distillation we fix $\tau = 1$ and $\alpha = \frac{1}{2}$ for all experiments. However, in Fig. 8 we show the mean Cohen's kappa for five repetitions of distillation with different choices of temperature τ, where the teacher model consist of 5 baseline models and is identical across all experiments. We let $\mathcal{D}_{\text{pseudo}}$ be all unscored data (120 nights) and let $\mathcal{D}_{\text{gt}}$ be the 15 scored training subjects, irrespective of subject type (long/short). We see that for $\tau \leq 2$ we observe an increase in Cohen's kappa, but for τ between 0.1 and 1 the differences are small. Finally, we also observe that using hard pseudo labels (i.e. one-hot encoded pseudo-labels with 1 at the entry of the largest class-probability) yields comparable performance to the best choices of τ, and can be a simple alternative to soft labels. This observation suggest that it is the utilization of additional data that is the key to the improved performance by distillation, rather than the implicit properties of the soft labels.

A.3. Discarding manually scored labels for distillation

During the distillation part of our training, we have the option to completely discard all manually obtained labels, and merely rely on the pseudo-labels produced by the teacher model. In Fig. 9 we show the performance (across all subjects) of student models trained with pseudo-labels instead of the ground-truth labels on the nights which were scored manually. Thus, we discard the manual labels, and effectively the distillation procedure is now fully unsupervised. However, these models slightly under-perform the models trained using the original manually produced labels. Furthermore, from Fig. 9 we also observe that without any additional data, simply using the soft labels is not sufficient to improve model predictive performance.

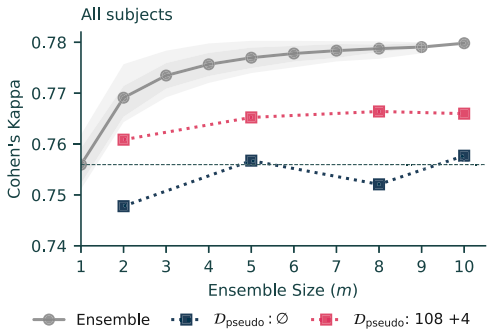


Fig. 9. Performance of students on all subjects, when the original labels of the scored nights are discarded and pseudo-labels are used on this dataset instead. We generally observe a slightly worse performance than when the original labels are used.

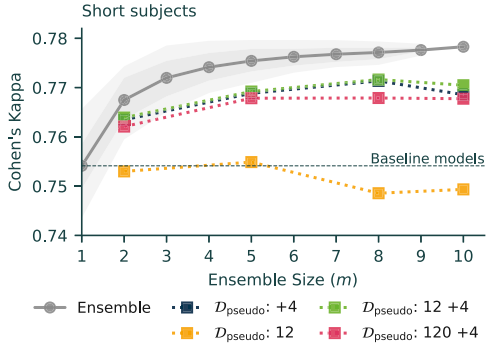


Fig. 10. Performance of personalized students on the short subjects in the same experiments as in Fig. 4(b). Note that these subjects do not have the additional 12 unscored nights, and any improvement observed here must be attributed to the use of pseudo-labeled test nights.

A.4. Performance on short subjects

In Section 3.2.2 and Fig. 4(b) we presented the predictive performance of personalized student models on *long subjects* when trained with unlabeled data from these test subjects. In Fig. 10 we report the performance on the *short subjects* in the exact same experiments. The lack of improvement by the yellow line is due to the fact that these models are trained identically to the baseline models, and we expect them to perform equally. However, we also observe an higher relative improvement in performance than for the long subjects as long as the unlabeled data contains the 4 test nights as unlabeled training data.

References

[1] Richard B. Berry, et al., AASM scoring manual updates for 2017 (version 2.4), J. Clin. Sleep Med. 13 (5) (2017) 665–666, http://dx.doi.org/10.5664/jcsm.6576.

[2] Merel M. van Gilst, et al., Protocol of the SOMNIA project: An observational study to create a neurophysiological database for advanced clinical sleep monitoring, BMJ Open 9 (11) (2019) e030996, http://dx.doi.org/10.1136/bmjopen-2019-030996.

[3] Pierrick J. Arnal, et al., The Dreem Headband as an Alternative to Polysomnography for EEG Signal Acquisition and Sleep Staging, BioRxiv (2019) 662734, http://dx.doi.org/10.1101/662734.

[4] Kaare B. Mikkelsen, et al., Accurate whole-night sleep monitoring with dry-contact ear-EEG, Sci. Rep. 9 (1) (2019) 1–12, http://dx.doi.org/10.1038/s41598-019-53115-3.

[5] Sirin W. Gangstad, et al., Automatic sleep stage classification based on subcutaneous EEG in patients with epilepsy, BioMed. Eng. OnLine 18 (1) (2019) 106, http://dx.doi.org/10.1186/s12938-019-0725-3.

[6] Kaare B. Mikkelsen, et al., Machine-learning-derived sleep–wake staging from around-the-ear electroencephalogram outperforms manual scoring and actigraphy, J. Sleep Res. (2018) e12786, http://dx.doi.org/10.1111/jsr.12786.

[7] Tomi Miettinen, et al., Success Rate and Technical Quality of Home Polysomnography With Self-Applicable Electrode Set in Subjects With Possible Sleep Bruxism, IEEE J. Biomed. Health Inf. 22 (4) (2018) 1124–1132, http://dx.doi.org/10.1109/JBHI.2017.2741522.

[8] Huy Phan, et al., SeqSleepNet: End-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging, (ISSN: 23318422) 2018, pp. 400–410, arXiv 27 (3).

[9] Jens B. Stephansen, et al., Neural network analysis of sleep stages enables efficient diagnosis of narcolepsy, Nature Commun. 9 (1) (2018) 5229, http://dx.doi.org/10.1038/s41467-018-07229-3.

[10] B. Koley, D. Dey, An ensemble system for automatic sleep stage classification using single channel EEG signal, Comput. Biol. Med. 42 (12) (2012) 1186–1195.

[11] Reza Boostani, et al., A comparative review on sleep stage classification methods in patients and healthy individuals, Comput. Methods Programs Biomed. 140 (2017) 77–91, http://dx.doi.org/10.1016/j.cmpb.2016.12.004.

[12] Huy Phan, Kaare Mikkelsen, Automatic sleep staging of EEG signals: Recent development, challenges, and future directions, Physiol. Meas. 43 (4) (2022) 04TR01, http://dx.doi.org/10.1088/1361-6579/ac6049.

[13] Kaare B. Mikkelsen, Simon L. Kappel, Danilo P. Mandic, Preben Kidmose, EEG Recorded from the Ear: Characterizing the Ear-EEG Method, Front. Neurosci. 9 (2015) http://dx.doi.org/10.3389/fnins.2015.00438.

[14] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, Distilling the Knowledge in a Neural Network, 2015, pp. 1–9, URL http://arxiv.org/abs/1503.02531.

[15] Lei Jimmy Ba, Rich Caruana, Do Deep Nets Really Need to be Deep? Adv. Neural Inf. Process. Syst. 3 (January) (2014) 2654–2662.

[16] Cristian Bucila, Rich Caruana, Alexandru Niculescu-Mizil, Model Compression, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '06, ACM, New York, NY, USA, 2006, pp. 535–541, http://dx.doi.org/10.1145/1150402.1150464.

[17] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, Yoshua Bengio, FitNets: Hints for thin deep nets, in: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 2015, pp. 1–13.

[18] Sergey Zagoruyko, Nikos Komodakis, Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer, in: 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, 2019, pp. 1–13.

[19] Suraj Srinivas, François Fleuret, Knowledge transfer with jacobian matching, in: 35th International Conference on Machine Learning, ICML 2018, vol. 11, 2018, pp. 7515–7523.

[20] Wonpyo Park, Dongju Kim, Yan Lu, Minsu Cho, Relational Knowledge Distillation, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019, http://dx.doi.org/10.1109/CVPR.2019.00409.

[21] J. Yim, D. Joo, J. Bae, J. Kim, A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 7130–7138, http://dx.doi.org/10.1109/CVPR.2017.754.

[22] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, Anima Anandkumar, Born Again Neural Networks, in: Jennifer Dy, Andreas Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 80, PMLR, Stockholmsmässan, Stockholm Sweden, 2018, pp. 1607–1616.

[23] Sungsoo Ahn, et al., Variational information distillation for knowledge transfer, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019-June, 2019, pp. 9155–9163, http://dx.doi.org/10.1109/CVPR.2019.00938.

[24] Yonglong Tian, Dilip Krishnan, Phillip Isola, Contrastive Representation Distillation, in: International Conference on Learning Representations, 2020, pp. 1–19.

[25] Raphael Gontijo Lopes, Stefano Fenu, Thad Starner, Data-Free Knowledge Distillation for Deep Neural Networks, 2017, arXiv preprint arXiv:1710.07535.

[26] Paul Micaelli, Amos J. Storkey, Zero-shot Knowledge Transfer via Adversarial Belief Matching, in: H Wallach, H Larochelle, A Beygelzimer, F d\textquotesingle Alché-Buc, E Fox, R Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019, pp. 9551–9561.

[27] Gongfan Fang, et al., Data-Free Adversarial Distillation, 2019, arXiv preprint arXiv:1912.11006.

[28] Rohan Anil, et al., Large scale distributed neural network training through online distillation, 2018.

[29] Mathilde Caron, et al., Emerging Properties in Self-Supervised Vision Transformers, 2021, arXiv preprint arXiv:2104.14294.

[30] Hossein Mobahi, Mehrdad Farajtabar, Peter L. Bartlett, Self-Distillation Amplifies Regularization in Hilbert Space, Adv. Neural Inf. Process. Syst. (2020).

[31] Kenneth Borup, Lars N. Andersen, Even your Teacher Needs Guidance: Ground-Truth Targets Dampen Regularization Imposed by Self-Distillation, in: M Ranzato, A Beygelzimer, Y Dauphin, P S Liang, J Wortman Vaughan (Eds.), Advances in Neural Information Processing Systems, vol. 34, Curran Associates, Inc., 2021, pp. 5316–5327.

[32] Kaare Mikkelsen, et al., Sleep monitoring using ear-centered setups: Investigating the influence from electrode configurations, IEEE Trans. Bio-Med. Eng. PP (2021) http://dx.doi.org/10.1109/TBME.2021.3116274.

[33] Kaare B. Mikkelsen, et al., Self-applied ear-EEG for sleep monitoring at home, in: 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2022, pp. 3135–3138, http://dx.doi.org/10.1109/EMBC48229.2022.9871076.

[34] Jacob Cohen, A Coefficient of Agreement for Nominal Scales, Educ. Psychol. Meas. 20 (1) (1960) 37–46, http://dx.doi.org/10.1177/001316446002000104.

[35] Pavel Izmailov, et al., Averaging weights leads to wider optima and better generalization, in: 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018, vol. 2, 2018, pp. 876–885.

[36] Timur Garipov, et al., Loss surfaces, mode connectivity, and fast ensembling of DNNs, Adv. Neural Inf. Process. Syst. 2018-Decem (Nips) (2018) 8789–8798.