Personalized Sleep State Classification via Learned Factor Graphs

Bar Rubinstein, Yoav Filin, Nir Shlezinger, and Nariman Farsad

***Abstract*—Recent years have witnessed a growing interest in using deep learning for sleep state tracking. A key challenge in doing so stems from the variability between different patients, and a model trained using the data of several patients may perform poorly on another subject. In this work, we study mechanisms for achieving personalized sleep state tracking, which can be utilized with various deep neural network (DNN) architectures. Our design uses learned factor graphs to exploit temporal correlation in a principled manner. Inspired by recent advances in federated learning, we incorporate schemes based on data and model interpolation for achieving personalized models. Our experimental study demonstrates that this approach achieves accurate classification with compact DNNs.**

***Index Terms*—** Sleep state monitoring, deep learning.

I. Introduction

Sleep disorders are known to affect a large portion of the general population [1]. The diagnosis and treatment of sleep disorders highly relies on the ability to accurately monitor the sleep stages of a patient, which change over time [2]. Classification into sleep states is often carried out manually based on the monitoring of multiple bio-electrical signals, and particularly using electroencephalography (EEG), electrooculogram (EOG), and submental electromyogram (EMG). This procedure is quite cumbersome and relies on human experts.

The unprecedented success of deep learning in the areas of computer vision and natural language processing gave rise to a growing research attention in its usage for sleep state tracking. The model-agnostic nature of deep neural networks (DNNs) and their ability to learn complex mappings indicate that they can facilitate automatic sleep stage tracking, relieving the dependence on human experts, while reducing the amount of monitored signals. Of particular interest is the task of EEG-based sleep state classification, where various DNN architectures utilizing convolutional networks [2]–[6], recurrent neural networks [7], and attention mechanisms [8], were proposed in the literature, as also surveyed in [9].

One of the key challenges in data-driven sleep state classifications stems from the high inter-subject variability of EEG signals [9]. DNNs trained using measurements acquired from a set of patients may thus perform poorly on a different subject. This can be tackled by manually acquiring measurements from the specific subject to be used for training. In fact, one can envision a patient going through a first stage of supervised monitoring in lab conditions, where personal measurements are acquired, followed by long-term automatic monitoring using lightweight, possibly portable, sleep state tracker. This motivates the proposal of mechanisms for training personalized data-driven systems, in a manner suitable for lightweight DNNs, which is the purpose of the current work.

In this work, we propose a framework for designing personalized DNN-based sleep state classifiers, that can be combined with various DNN architectures. Our design is based on identifying a set of established model assumptions on the expected behavior of EEG signals and the latent sleep state. We build upon the Markovian model for temporal evolution, utilizing the DNN to compute the function nodes of the factor graph [10]–[13], rather than as a classifier. The learned factor graph is combined with belief propagation (BP) inference to exploit temporal correlation in a principled manner.

Then, we tackle the degrading effects of inter-subject variability. This is achieved by extending the learned function nodes into a deep ensemble [14], which improves accuracy and facilitates distributed implementation. We propose methods for training based on a large collective dataset from other subjects as well as a small personal dataset. We adapt personalizing concepts recently proposed for federated learning [15]: The first approach trains the ensemble using the complete dataset; the remaining two support reuse of an ensemble trained with the collective dataset by either including an additional personal model, or by learning to combine the ensemble as a mixture-of-experts. Our numerical experiments, which use the PhysioNet Sleep-EDF database [16], demonstrate that the proposed combination allows achieving accurate recovery of over 88% with 90% detection for four out of the five sleep states, while utilizing a simple compact DNN architecture.

The rest of this paper is organized as follows; Section II details the system model. Section III presents the proposed sleep state tracking system. Experimental results are reported in Section IV, and Section V provides concluding remarks.

II. System Model

A. Problem Formulation

We consider sleep pattern tracking from EEG measurements. The measured EEG signal of a patient is divided into subsequent non-overlapping segments, referred to as epochs [17]. The input is thus a multivariate time sequence, denoted $\{\boldsymbol{y}_n^j\}_{n=1}^{N_j}$ for the jth patient, where $\boldsymbol{y}_n^j$ represents the nth epoch and has K features. Here, N_j denotes the number of recorded epochs, which can change between patients.

The measured EEG signal is related to the sleep state of the patient, which changes in time along the recording. We consider five stages of sleep: awake (AWA), REM, and non-REM sleep stages (N1-N3). Assuming that each EEG epoch corresponds to a single sleep state, we represent the sleep state of the jth patient as a time sequence $\{s_n^j\}_{n=1}^{N_j}$, where each s_n^j takes values in $\mathcal{S} := \{\text{Wake}, \text{REM}, \text{N1}, \text{N2}, \text{N3}\}$.

Our goal is to design a personalized sleep state tracking system. Namely, for a patient of index k, we aim to find a mapping that recovers the sleep states from $\{\boldsymbol{y}_n^k\}_{n=1}^{N_k}$. To design the personalized mapping, we utilize data divided into two sets:

- *Collective data* - a large set of EEG measurements and their corresponding sleep states taken from a set of patients $\mathcal{J}$ which does not include the current patient k, i.e., $\{\{(\boldsymbol{y}_n^j, s_n^j)\}_{n=1}^{N_j}\}_{j\in\mathcal{J}}$;
- *Personal data* - a small set of EEG measurements and sleep states taken from the kth patient comprised of $\tilde{N}_k < N_k$ epochs, i.e., $\{(\boldsymbol{y}_n^k, s_n^k)\}_{n=1}^{\tilde{N}_k}$.

B. Rubinstein, Y. Filin, and N. Shlezinger are with the School of ECE, Ben-Gurion University of the Negev, Beer-Sheva, Israel (email: {barrub, yoavfi}@post.bgu.ac.il, nirshl@bgu.ac.il). N. Farsad is with the CS Department, Ryerson University, Toronto, ON (e-mail: nfarsad@ryerson.ca).

B. Model Assumptions

To design personalized sleep state tracking, we introduce four assumptions, which are utilized in the systems proposed in Section III. These assumptions are based on accumulated knowledge and established approximations regarding EEG signals and their relationship with the sleep pattern.

A1 The statistical model relating the EEG signals and the sleep states varies between patients. Thus, a tracking system suitable for one patient may not be suitable for another.

A2 The statistical model relating the EEG measurements and the state is quite complex and difficult to express analytically.

A3 Each EEG epoch is statistically related to its sleep state, which is a random function of the previous state. Thus, the joint distribution of the sleep states and the EEG measurements obeys a Markovian model, i.e., for each patient j

$$P(\{s_n^j, \boldsymbol{y}_n^j\}_{n=1}^{N_j}) = \prod_{n=1}^{N_j} P\left(\boldsymbol{y}_n^j | s_n^j\right) P\big(s_n^j | s_{n-1}^j\big). \tag{1}$$

A4 The number of epochs in each EEG signal, i.e., the length of the considered time sequences, is not fixed, and can vary considerably between different patients and recordings.

III. Sleep State Tracking

In this section we introduce the proposed personalized sleep state tracking system. Our design builds upon the model assumptions *A1*-*A4*. The Markovian approximation *A3* along with the varying sequence length *A4* motivate the usage of *factor graph methods*, which facilitate inference over factorizable distributions of varying length [18]. Accounting for the intractability of the model *A2*, leads us to utilizing learned factor graphs, described in Subsection III-A. We tackle the statistical diversity by combining ensemble models with personalization mechanisms in Subsection III-B, and provide a discussion in Subsection III-C.

A. Learned Factor Graphs

Factor graph methods facilitate inference from factorizable distributions by message passing over a graphical representation of the distribution known as a *factor graph* [19]. For the Markovian model (1), the maximum a-posteriori probability (MAP) rule, whose complexity grows exponentially with the sequence length, is computed via message passing with complexity that grows linearly with N_j.

To see this, we define the *function nodes*

$$f_n^j(\boldsymbol{y}_n^j, s_n^j, s_{n-1}^j) := P\left(\boldsymbol{y}_n^j | s_n^j\right) P\big(s_n^j | s_{n-1}^j\big), \tag{2}$$

which can be used to form a graphical representation of the statistical dependencies as a factor graph [19]. We can now write the MAP rule $\hat{s}_n^j = \arg\max_{s_n \in \mathcal{S}}(s_n^j, \{\boldsymbol{y}_n^j\})$ as

$$\hat{s}_n^j = \arg\max_{s_n \in \mathcal{S}} \overrightarrow{\mu}_n^j(s_n) \overleftarrow{\mu}_n^j(s_n), \tag{3}$$

where $\overrightarrow{\mu}_n^j(s_n)$ and $\overleftarrow{\mu}_n^j(s_n)$ are the *forward* and *backward* messages, respectively. These messages are defined as $\overrightarrow{\mu}_n^j(s_n) := \big(\sum_{s_1^j,\ldots s_{n-1}^j} \prod_{i=1}^n f_i^j(\boldsymbol{y}_i^j, s_i^j, s_{i-1}^j)\big)$, and $\overleftarrow{\mu}_n^j(s_n) := \big(\sum_{s_{n+1}^j,\ldots s_{N_j}^j} \prod_{i=n+1}^{N_j} f_i^j(\boldsymbol{y}_i^j, s_i^j, s_{i-1}^j)\big)$, and are computed recursively. This inference method (3) coincides with the BP algorithm for hidden Markov models [19].

BP (3) computes the MAP rule for statistical models obeying (1) in a manner which is invariant of the sequence length and with complexity that only grows linearly with it. This makes factor graph methods suitable for the problem based on assumptions *A3*-*A4*. Nonetheless, to compute (3), one must evaluate the function nodes (2), which may be intractable *A2*. Here, we exploit the presence of data to learn the function nodes, as proposed in [10], using dedicated DNNs.We note that the function nodes defined in (2) are comprised of two terms: The state transition probability $P\big(s_n^j | s_{n-1}^j\big)$ which can be represented using an $|\mathcal{S}| \times |\mathcal{S}| = 5 \times 5$ matrix. This quantity is estimated once via histogram; and the observations model $P\left(\boldsymbol{y}_n^j | s_n^j\right)$, which maps an observation $\boldsymbol{y}_n^j$ into $|\mathcal{S}|$ different values for each possible state. This mapping is modeled as a classification DNN with input $\boldsymbol{y}_n^j$ and $|\mathcal{S}| = 5$ categories, that is learned from data to minimize the cross-entropy loss. Finally, we use the same learned model for each function node. Namely the same DNN and histogram are reused to estimate $f_n^j(\boldsymbol{y}_n^j, s_n^j, s_{n-1}^j)$ for each observed $\boldsymbol{y}_n^j$.

B. Deep Personalized Function Nodes Ensemble

The usage of learned factor graphs detailed in the previous subsection is suitable for the problem of sleep state tracking in light of assumptions *A2*-*A4*, and previous applicatoins of this methodology in other applications involving bio-medical data in [12], [13]. However, the design does not specify which parametric model one must use to learn the function nodes, i.e., which DNN should be used to estimate the observation model. Here, we focus on the operation of the learned function nodes, incorporating deep ensembles and methods for exploiting the collective and personal data to achieve personalized models.

1) Ensemble Function Nodes: To handle the heterogeneity among different users *A1*, we design the learned function nodes utilizing deep ensembles. Deep ensembles are scalable architectures comprised of multiple diverse DNNs, which infer by aggregating the individual predictions. Deep ensembles were shown to achieve high accuracy and generalization performance in a manner that improves with the number of individual DNNs, while being relatively robust to statistical heterogeneity [14], [20], [21].

We train $E > 0$ diverse DNN classifiers, and combine their outputs to estimate the observations conditional distribution [20]. Various strategies were proposed to achieve diverse DNNs to form an ensemble, including data bagging [14] and regularization [20]. Here, we use a different random initialization for each DNN while reusing the same data for training [21]. An illustration is depicted in Fig. 1(a).

2) Personalization of Sleep State Classifier: We study three approaches to utilize the collective and personal datasets, detailed in Subsection II-A, to yield a personalized inference rule using deep function node ensembles:

Presonalized Ensemble: The direct approach uses the complete dataset, both collective and personal, to train the ensemble. The advantage of this approach, also referred to as *data interpolation* [15], is that all models in the ensembles are trained using a relatively large dataset, out of which some portion reflects on the distribution of the considered patient. This approach is expected to yield the highest accuracy, as numerically observed in Section IV. The main drawback is that the complete ensemble is trained anew for each inspected patient in order to yield a personalized inference rule.

Modular Ensemble: This method uses the larger collective dataset to train the ensemble, and the smaller personal dataset to train a distinct classifier. During inference, the personal model is added to the ensemble to boost personalized function node computations. The main advantage of this approach is its modularity; one can train a single collective ensemble that is reused for multiple patients by combining with their personal trained model, as proposed in the context of federated learning [15], [22]. This is illustrated in Fig. 1(b).

Mixture-of-Experts: An alternative modular approach that supports reuse of a learned collective ensemble replaces the averaging of the models with a personalized learned combining. Here, each model in the ensemble is trained using the collective data set, allowing the ensemble to be reused among multiple patients. Then, for each patient, the personal dataset is utilized to learn to combine the outputs as a form of mixture-of-experts, as illustrated in Fig. 1(c).

C. Discussion

The proposed sleep state classifier is comprised of three main components. The first is the usage of learned factor graphs to

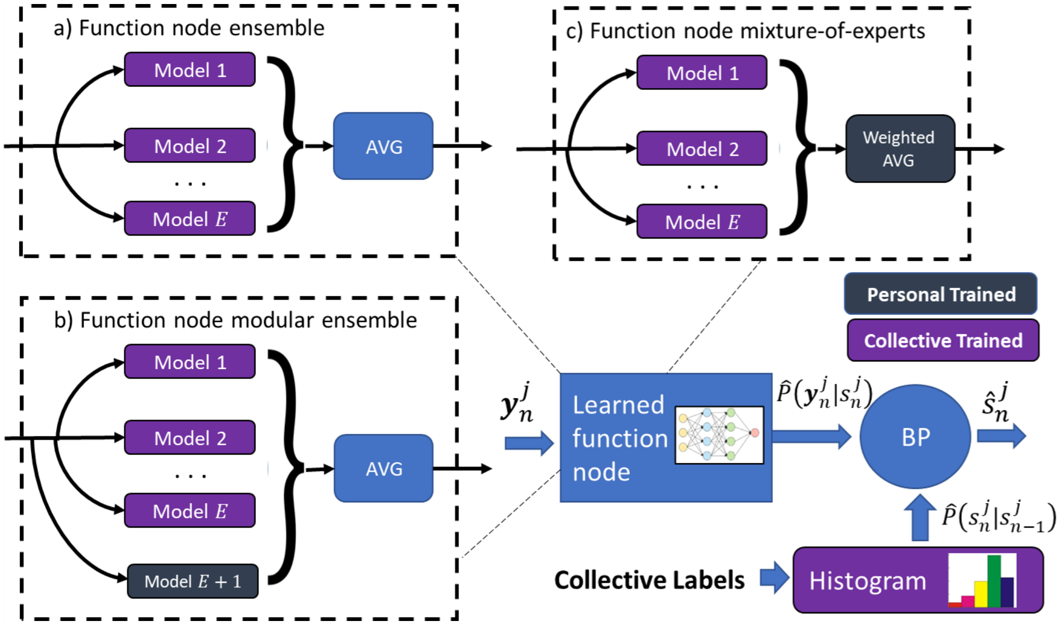


Fig. 1: Learned function models illustrations.

exploit the temporal correlation of the hidden sleep state process in a principled manner, that can be combined with various DNN architectures, using their output as an estimate of the function nodes rather than as classifiers. The second component is the usage of deep ensembles, which yield improved accuracy, particularly in the presence of statistically heterogeneous data. These components allow achieving accurate classification while utilizing lightweight DNNs, as we numerically demonstrate in Section IV.

On top of the deep ensembles, we incorporate mechanisms for personalization inspired by recent advances in federated learning, where one also often deals with multiple models and non-i.i.d. data. The personalized ensemble approach is based on *data interpolation*, which personalizes by merging data sets, while the modular ensemble and mixture-of-experts schemes follow a *model interpolation* approach, where models trained on different data sets are combined for personalized inference [15]. In our numerical study we show that personalized ensembles typically achieve the best accuracy among the approaches, inline with similar findings in [15]. Nonetheless, model interpolation may be preferred in some cases due to its modularity, which allow a trained ensemble to be reused for multiple systems.

IV. Experimental Study

A. Experimental Setup

Our experimental study uses the PhysioNet Sleep-EDF Expanded database [16]. This dataset consists of 197 whole-night PolySomnoGraphic sleep recordings, containing EEG, EOG, chin EMG, and event markers. As in [5], [7], we use 20 patients from one of the two studies in this dataset that investigates the age effect in healthy subjects, known as the Sleep Cassette (SC) dataset. We focus on using a single EEG channel recording (channel Fpz-Cz), and apply the feature extraction method proposed in [17] to the EEG epochs. As a result, every segment of 30 seconds of recording are represented using 150 features, i.e., $K = 150$. For each inspected patient of index k, we use the data of the remaining patients as the collective dataset; the first 80% of the segments of patient k are the personal dataset; and the rest are used for test.

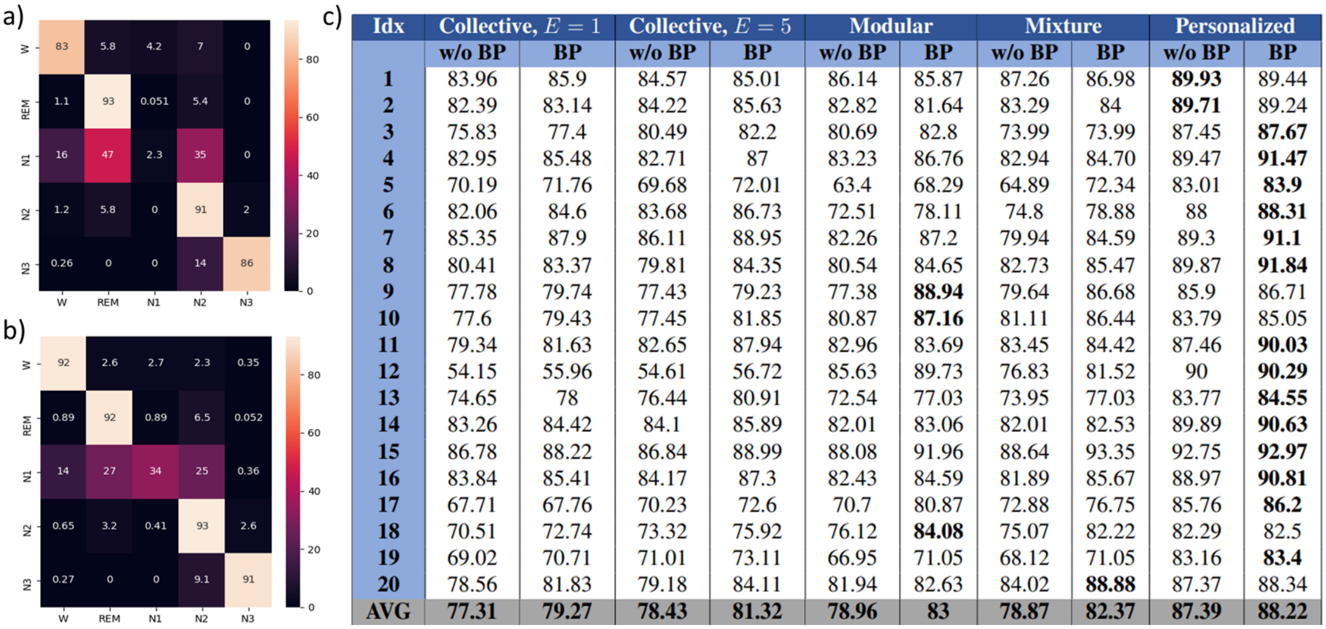
The basic DNN is a 5-layer fully-connected network. The hidden layers are of sizes $1500, 800, 250, 100$, and 5, with intermediate ReLU activations and a softmax output layer. The network is trained using stochastic gradient descent with learning rate 10^{-3}, weight decay 10^{-5}, and momentum 0.9, over 600 epochs with a mini-batch size of 64 samples. The relatively simple architecture is used to capture the gains of each of the considered mechanisms, i.e., learned factor graphs, ensemble function nodes, and the personalization schemes. This DNN is used by the following systems:

- *Collective:* Here, E DNNs are trained using the collective dataset, as in the leave-one-out approach [7]. We use both $E = 1$ and $E = 5$ to quantify the gains of using ensembles with output averaging compared to a single model.
- *Modular:* a modular ensemble comprised of $E = 5$ collective DNNs and a single personalized DNN.
- - *Mixture:* a mixture-of-experts model with $E = 5$ models trained on the collective dataset, combined using a weighted average learned using the personal dataset.
- *Personalized:* An ensemble of $E = 5$ DNNs trained using the complete dataset (both collective and personalized).

Each of these systems is used both as a classifier, namely, to recover s_n^k from the observed $\boldsymbol{y}_n^k$, as well as to estimate the conditional distribution utilized for BP inference. For the latter, we use the collective data to estimate the state transition probability via histogram. Setting $E = 5$ was based on empirical trials, where increasing E while training as in [21] was observed to hardly affect accuracy, as also noted in [23].

B. Results

The results are summarized in Fig 2(c). We observe that the usage of learned factor graphs , i.e., incorporating the model output into BP inference rather than for direct classification, yields a consistent improvement, which ranges from 0.83% to 4.1%. It is also observed that utilizing ensemble models results in improved accuracy, where the accuracy of the collective configuration increases by $1.1\%-2.1\%$. Furthermore, we note that incorporating the personalized data allows improves accuracy compared to using solely the collective dataset. This benefit, which is exhibited by all considered methods, is most notable when observing the accuracy for patient index $k = 12$; here, the collective models achieve at most 56% accuracy, while the personalized systems achieve accuracy which varies from 76% (mixture without BP) up to 90.3% (personalized with BP). Among the proposed methods for utilizing the personal dataset, personalized



a)

	W	REM	N1	N2	N3
W	83	5.8	4.2	7	0
REM	1.1	93	0.051	5.4	0
N1	16	47	2.3	35	0
N2	1.2	5.8	0	91	2
N3	0.26	0	0	14	86

b)

	W	REM	N1	N2	N3
W	92	2.6	2.7	2.3	0.35
REM	0.89	92	0.89	6.5	0.052
N1	14	27	34	25	0.36
N2	0.65	3.2	0.41	93	2.6
N3	0.27	0	0	9.1	91

c)

Idx	**Collective, $E = 1$**		**Collective, $E = 5$**		**Modular**		**Mixture**		**Personalized**	
	w/o BP	**BP**	**w/o BP**	**BP**	**w/o BP**	**BP**	**w/o BP**	**BP**	**w/o BP**	**BP**
1	83.96	85.9	84.57	85.01	86.14	85.87	87.26	86.98	**89.93**	89.44
2	82.39	83.14	84.22	85.63	82.82	81.64	83.29	84	**89.71**	89.24
3	75.83	77.4	80.49	82.2	80.69	82.8	73.99	73.99	87.45	**87.67**
4	82.95	85.48	82.71	87	83.23	86.76	82.94	84.70	89.47	**91.47**
5	70.19	71.76	69.68	72.01	63.4	68.29	64.89	72.34	83.01	**83.9**
6	82.06	84.6	83.68	86.73	72.51	78.11	74.8	78.88	88	**88.31**
7	85.35	87.9	86.11	88.95	82.26	87.2	79.94	84.59	89.3	**91.1**
8	80.41	83.37	79.81	84.35	80.54	84.65	82.73	85.47	89.87	**91.84**
9	77.78	79.74	77.43	79.23	77.38	**88.94**	79.64	86.68	85.9	86.71
10	77.6	79.43	77.45	81.85	80.87	**87.16**	81.11	86.44	83.79	85.05
11	79.34	81.63	82.65	87.94	82.96	83.69	83.45	84.42	87.46	**90.03**
12	54.15	55.96	54.61	56.72	85.63	89.73	76.83	81.52	90	**90.29**
13	74.65	78	76.44	80.91	72.54	77.03	73.95	77.03	83.77	**84.55**
14	83.26	84.42	84.1	85.89	82.01	83.06	82.01	82.53	89.89	**90.63**
15	86.78	88.22	86.84	88.99	88.08	91.96	88.64	93.35	92.75	**92.97**
16	83.84	85.41	84.17	87.3	82.43	84.59	81.89	85.67	88.97	**90.81**
17	67.71	67.76	70.23	72.6	70.7	80.87	72.88	76.75	85.76	**86.2**
18	70.51	72.74	73.32	75.92	76.12	**84.08**	75.07	82.22	82.29	82.5
19	69.02	70.71	71.01	73.11	66.95	71.05	68.12	71.05	83.16	**83.4**
20	78.56	81.83	79.18	84.11	81.94	82.63	84.02	**88.88**	87.37	88.34
AVG	**77.31**	**79.27**	**78.43**	**81.32**	**78.96**	**83**	**78.87**	**82.37**	**87.39**	**88.22**

Fig. 2: Confusion matrices for (a) mixture ensemble with BP and (b) personalized ensemble with BP; (c) Accuracy comparison.

deep ensembles achieves the best performance, with accuracy of over 90% for 8 out of the 20 patients, and 88.2% on average over all patients. Modular ensembles combined with BP achieves comparable accuracy, with an average of 83%, while enabling to reuse the collective ensemble among multiple patients.

Finally, as the sleep states are typically non-balanced, the accuracy measure does not fully capture the performance. We thus evaluate the confusion matrices for mixture ensemble with BP and personalized ensemble with BP in Fig. 2(a)-(b), respectively. We observe that for these leading models, a large portion of the errors reported in Fig. 2(c) corresponds to confusing state N1, as also noted in [2]. The personalized ensemble model is shown to correctly identify all states except for N1 with accuracy of over 91%, while using a relatively simple DNN. This indicates the improvements one can achieve by combining patient-specific data with collective datasets. Doing so allows to realize personalized sleep state tracking systems using designs exploiting domain knowledge and established model assumptions.

V. Conclusions

In this work we studied methods for designing DNN-based personalized sleep state classifiers. We identified established assumptions that capture some of the key properties of such setups, which were used as domain knowledge in our design. We proposed inference based on data-driven factor graphs combining deep ensembles with personalized data. Our numerical results demonstrate the gains of each of the components of our design, where the proposed personalized ensemble with BP achieves over 90% for 8 out of 20 patients while utilizing compact DNNs.

References

[1] C. V. Senaratna, J. L. Perret, C. J. Lodge, A. J. Lowe, B. E. Campbell, M. C. Matheson, G. S. Hamilton, and S. C. Dharmage, "Prevalence of obstructive sleep apnea in the general population: a systematic review," *Sleep Medicine Reviews*, vol. 34, pp. 70–81, 2017.

[2] H. Korkalainen, J. Aakko, S. Nikkonen, S. Kainulainen, A. Leino, B. Duce, I. O. Afara, S. Myllymaa, J. Töyräs, and T. Leppänen, "Accurate deep learning-based sleep staging in a clinical population with suspected obstructive sleep apnea," *IEEE J. Biomed. Health Inform.*, vol. 24, no. 7, pp. 2073–2081, 2019.

[3] O. Tsinalis, P. M. Matthews, Y. Guo, and S. Zafeiriou, "Automatic sleep stage scoring with single-channel EEG using convolutional neural networks," *arXiv preprint arXiv:1610.01683*, 2016.

[4] S. Chambon, M. N. Galtier, P. J. Arnal, G. Wainrib, and A. Gramfort, "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 4, pp. 758–769, 2018.

[5] A. I. Humayun, A. S. Sushmit, T. Hasan, and M. I. H. Bhuiyan, "End-to-end sleep staging with raw single channel EEG using deep residual convnets," in *Proc. IEEE BHI*, 2019.

[6] W. Neng, J. Lu, and L. Xu, "CCRRSleepNet: A hybrid relational inductive biases network for automatic sleep stage classification on raw single-channel EEG," *Brain Sciences*, vol. 11, no. 4, p. 456, 2021.

[7] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "SeqSleepNet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 3, pp. 400–410, 2019.

[8] W. Qu, Z. Wang, H. Hong, Z. Chi, D. D. Feng, R. Grunstein, and C. Gordon, "A residual based attention model for eeg based sleep staging," *IEEE J. Biomed. Health Inform.*, vol. 24, no. 10, pp. 2833–2843, 2020.

[9] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalography analysis: a systematic reviewv," *Journal of neural engineering*, vol. 16, no. 5, p. 051001, 2019.

[10] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "Learned factor graphs for inference from stationary time sequences," *IEEE Trans. Signal Process.*, vol. 70, pp. 366–380, 2021.

[11] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," *arXiv preprint arXiv:2012.08405*, 2020.

[12] B. Salafian, E. F. Ben-Knaan, N. Shlezinger, S. Ribaupierre, and N. Farsad, "CNN-aided factor graphs with estimated mutual information features for seizure detection," in *Proc. IEEE ICASSP*, 2022.

[13] E. F. Ben-Knaan, Y. C. Eldar, and N. Shlezinger, "Recovery of noisy pooled tests via learned factor graphs with application to COVID-19 testing," in *Proc. IEEE ICASSP*, 2022.

[14] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.

[15] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," *arXiv preprint arXiv:2002.10619*, 2020.

[16] B. Kemp, "The sleep-EDF database online," 2013.

[17] D. Jiang, Y.-n. Lu, M. Yu, and W. Yuanyuan, "Robust sleep stage classification with single-channel EEG signals using multimodal decom-

position and HMM-based refinement," *Expert Systems with Applications*, vol. 121, pp. 188–203, 2019.

[18] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proc. IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007.

[19] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, 2004.

[20] B. Brazowski and E. Schneidman, "Collective learning by ensembles of altruistic diversifying neural networks," *arXiv preprint arXiv:2006.11671*, 2020.

[21] S. Fort, H. Hu, and B. Lakshminarayanan, "Deep ensembles: A loss landscape perspective," *arXiv preprint arXiv:1912.02757*, 2019.

[22] E. L. Zec, O. Mogren, J. Martinsson, L. R. Sütfeld, and D. Gillblad, "Federated learning using a mixture of experts," *arXiv preprint arXiv:2010.02056*, 2020.

[23] N. Shlezinger, E. Farhan, H. Morgenstern, and Y. C. Eldar, "Collaborative inference via ensembles on the edge," in *Proc. IEEE ICASSP*, pp. 8478–8482, 2021.