Automated Sleep Staging Annotation via Shallow and Deep Learning
CSE 6250 Project Final Paper

Hemin Yang, Chenyang Shi
YouTube Presentation Link
College of Computing, Georgia Institute of Technology, 801 Atlantic Dr NW, Atlanta, GA 30332

Abstract

Sleep stage annotation as inferred from polysomnogram (PSG) signals is central in analyzing sleep data as it detects potential sleep disorders at an early stage. However, this task is still in the domain of an expert sleep technologist. Using open dataset of the Sleep Heart Health Study (SHHS), in this paper, we explored deploying a variety of basic machine learning and deep learning algorithms on classifying sleep stages based on raw electroencephalogram (EEG) signals. The models were trained on intra- and inter-subjects, respectively. For intra-subject study Recurrent Neural Network (RNN) achieved an overall accuracy of 87%, while for inter-subject, the best performance came from convolutional neural network (CNN) with a classification accuracy of 78%. A further performance enhancement was attempted with an ensemble approach, where a super learner method yielded an accuracy of 80% on inter-subject study. In both cases, deep learning outperform basic learning by a large margin.

1 Introduction

Sleep is central to human health. Sleep disorders such as sleep apnea, narcolepsy and hypersomnia, cataplexy, and sleeping sickness are world-wide health concerns, causing a wide range of deleterious health consequences including an increased risk of hypertension, diabetes, obesity, depression, heart attack, and stroke. According to Hillman et al.[1], about 50-70 million people in the United States alone are suffering from sleep disorders. The ability to detect sleep disruption from recorded sleep data at an early stage is therefore of great value. Based on the manual published by American Academy of Sleep Medicine (AASM)[2], sleep is categorized into five stages. They are the stage of Rapid Eye Movement (R), and three non-R stages (N1, N2, N3 where N3, also called Slow Wave Sleep, can be further divided into two stages, N3 and N4), and a Wake (W) stage. When studying human sleep behavior, electrical sensors are placed at different parts of the body. The entire electrical activities from these sensors are recorded in a polysomnogram (PSG) that includes an electroencephalogram (EEG), an electrooculogram (EOG), an electromyogram (EMG) and an electrocardiogram (ECG). After an overnight PSG signal is collected, it is divided into 30 seconds intervals, called `epochs`. A central part of analyzing the PSG signal is to classify each epoch into one of the five sleep stages (R, N1, N2, N3, and W). A manual annotation requires specialized training from expert sleep technologists and is thus expensive. Also, even for the experts, a manual scoring is laborious and time-consuming–it may take up to one hour from a technologist's time to generate a complete sleep report. An automatic annotation by computers is an important alternative approach which has seen recent progresses.[3–6]

In recent years, machine learning and deep learning algorithms have been applied to solve a wide range of problems in science and health industry. Compared with conventional shallow learning methods, the deep learning algorithm, that utilizes multiple layers of linear and non-linear processing units, has the advantage of learning hierarchical representations or features from large compiled sources of data. In this paper, we applied and compared a range of deep learning neural networks together with basic shallow learning algorithms to realize an automatic scoring of sleep stages based on open EEG datasets downloaded from Sleep Heart Health Study (SHHS)[7,8]. We investigated both intra- and inter-subjects scenarios where the former uses data from a single subject for training and testing, while the later has no such constraints (i.e. epochs data can be used across subjects). Our intra-subject results based on 100 patients' data suggest that all deep learning algorithms outperformed basic machine learning algorithms by a large margin ($>$0.3 in prediction accuracy). The best deep learning algorithm is Recurrent Neural Network (RNN), achieving a classification accuracy of 86%. As a comparison, the best shallow learning algorithm is multilayer perceptron (MLP) with an accuracy of 51%. For inter-subject study, convolutional neural network (CNN) gives the best classification accuracy of 78%. An enhanced performance of 80% was achieved by ensemble learning with a super learner.

Table 1: Recent progresses of annotating sleep stage using machine learning techniques.

Research groups	Data set	Machine learning algorithm	Best accuracy
Tsinalis et al.[4]	PhysioNet repository	CNN	74%
Supratak et al.[5]	MASS/Sleep-EDF	CNN	86.2% / 82.0%
Biswal et al.[3]	3.2 TB EEG data	LR/TB/MLP; CNN/RNN/RCNN	85.76%

2 Related Work

In the literature, Tsinalis et al.[4] attempted the sleep stage scoring problem by using convolutional neural networks (CNNs). They used an open dataset from PhysioNet repository on 20 healthy young adults. They trained CNNs with raw EEG signals without preprocessing. A class-balanced random sampling within the stochastic gradient descent (SGD) optimization of the CNN was used to avoid skewed performance in favor of the most represented sleep stages. They achieved a F1 score of 81%, mean accuracy of 82% and overall accuracy of 74%. The authors believe their results are comparable to state-of-art methods with hand-engineered features.

Supratak et al.[5] proposed a model named DeepSleepNet based on raw single-channel EEG data. Instead of hand-engineering features, they automated the process by utilizing the feature extraction capabilities of deep learning. In their CNNs architecture, they built two of them at first layers (to extract time-invariant features from raw single channel EEG) and bidirectional Long-Short Term Memory (to encode temporal information such as sleep stage transition rules into the model). They implemented a two-step training algorithm to train their model efficiently. The model was trained on two datasets MASS (Montreal Archive of Sleep Studies)[9] and Sleep-EDF[10,11], and achieved an overall accuracy of 86.2% and 82.0% (for MASS and Sleep-EDF) and macro F1 score of 81.7% and 76.9%. The work from the same research lab[6] applied LSTM on time-frequency domain features from the F4-EOG and Fp2-EOG channels separated. Although the authors are satisfied with the overall results, they are not happy with the fact that the features have been hand-engineered before feeding into neural networks.

Biswal et al.[3] have proposed SLEEPNET for annotation of sleep stage. They analyzed sleep data from 10,000 patients with a total 80,000 hours of recorded EEG data (3.2 TB data).They performed feature engineering on the EEG signals and extracted three types of features, i.e. raw EEG signals in the time domain, spectrograms in the frequency domain (after Fourier transforming the time signals) and expert defined features. Later, they applied a series of conventional machine learning (Logistic Regression, Tree Boosting, MultiLayer Perception) and deep learning algorithms (Convolutional Neural Network, Recurrent Neural Network, and Recurrent-Convolutional Neural Network) to attack the classification problems. Among these methods, Recurrent Neural Network was found to achieve the best performance with an average accuracy of 85.76%, comparable to the human-level annotation performance.

A summary of these literature reports is shown in Table 1.

3 System Description

Based on the literature research, we lay out our research problem as proposing a model that performs automatic sleep stage scoring based on EEG data, i.e., annotates the sleep stage of every given 30 second EEG signal epoch. The overall performance of our model, as will be assessed by the accuracy, should be comparable or better than that from sleep technologists and/or previously reported literature results.

3.1 Basic Machine Learning Algorithms

In this work we applied five basic machine learning algorithms to classify the sleep stages. The shallow learning algorithms explored include Decision Trees (DT), Random Forest (RF), Multi-Layer Perceptron (MLP), AdaBoost (AB) and Logistic Regression (LR).

DT is an intuitive learning method that predicts target values based on a set of decision rules inferred from the features. In the tree structures, the leaf nodes represent labels and branches represent conjunctions of features that lead to those labels. DT starts learning by splitting the entire dataset into subsets, and the process is repeated in a recursive manner until either the subset at a leaf node has the small value of the target variable or further splitting no longer adds value

to the predictions. First introduced by T. Ho in 1995[12], RF is an ensemble learning method for classification and regression problems. RF is learned by building a number of decision trees on various sub-samples of the dataset and using averaging to improve the predictive accuracy and combat over-fitting as easily seen in decision trees. MLP is a class of feed-forward artificial neuron network. In an MLP, the features from the input layers are fed into neural network defined by number of layers and number of neurons per layer. Since non-linear activation functions are used, MLP can learn complex relationship between features and labels. AB is one of the boosting methods that combine several weak learners into a strong learner. The general idea of most boosting methods is to train predictors sequentially, each trying to correct its predecessor. In AB a sequence of weak learners are fitted on repeated modified versions of the data. The final prediction is made based on a weighted majority vote. LR is a linear model for classification. It uses a logistic function to model the probabilities for the possible outcomes of a single trial.

3.2 Convolutional Neural Network

Convolutional neural network (CNN) or convnets is a special type of neural network that has been successful at computer vision tasks. Different from densely connected neuron layers that learn global patterns in their input feature space, the convnet is skilled at learning local patterns. In current study, the local spatial features embedded in raw EEG signals will be learned by CNN.

3.3 Recurrent Neural Network

Recurrent neural network (RNN) is designed to learn from sequential data. For classical neural networks, learning is memoryless which means the temporal order of events has nothing to do with learning. This is the major shortcoming of classical neural networks. For example, the sleep stage of previous 30 second epoch can be very related with the current epoch. RNN solves this problem by feeding back the output of one layer to its input, which allows information to be passed from one step of the network to the next.

3.4 Recurrent-Convolutional Neural Network

While RNN excels at learning sequential data, CNN is capable of learning data having spatial locality. Combining a RNN and CNN, we can get a hybrid model which can learn spatial features as well as preserve the long-term temporal relationship. In RCNN, raw features are first connected to a CNN, which will extract spatial features from EEG. These features are then fed into a RNN to learn the temporal dependency.

4 Experiments

We conducted both intra-subject and inter-subject experiments. In intra-subject experiments, epochs from one subject are separated into training and testing sets whereas in inter-subject case, all epochs from different subjects can be used for either training or testing. Intuitively, intra-subject experiments can achieve higher accuracy because epochs from the same subject are more relevant. Nonetheless, the inter-subject case is more practically useful in health care study because a trained model is more desired that can do automatic stage annotation for any new subject during the entire sleep.

4.1 Dataset Description

We use the EEG records from the open dataset of the Sleep Heart Health Study (SHHS) provided by National Sleep Research Resource (NSRR)[7,8] in this research. The SHHS dataset consists of two parts: SHHS1 and SHHS2. We will only use SHHS1 in this project. SHHS1 data are collected from 5,793 subjects from 1995-1998. The EEG signals were sampled at 125 Hz from C3-A2 and C4-A1 channels. And the raw data storage for these EEG records is about 313 GB. Besides the raw EEG records, the raw dataset also provides the sleep stage per 30-second epoch for every subject in XML format. The possible stages are 0, 1, 2, 3, 4, 5, 6, and 9, which indicate wake (W), stage 1 sleep (N1), stage 2 sleep (N2), stage 3 sleep (N3), stage 4 sleep (N4), and REM (R), movement, and unscored respectively. To compare with the results from Biswal et al.[3], we filter the samples with stage 6 and 9 because they are out of the scope of our study and treat stage 3 and 4 as the same stage (i.e., N3).

Table 2: Summary of the SHHS1 dataset

Number of subjects	Hours of EEG data	Number of epochs	N_W	N_{N1}	N_{N2}	N_{N3}	N_R
5,793	48,861	5,863,351	1,691,288	217,583	2,397,460	739,403	817,473

Note: N_W: number of W epochs, N_{N1}: number of N1 epochs, N_{N2}: number of N2 epochs, N_{N3}: number of N3 and N4 epochs, N_R: number of R epochs.

Table 3: Intra-Subject Performance of basic machine learning algorithms

	Hyperparameter	Search Space	Best-tuned value	Accuracy (This study)	Accuracy (Biswal et al.)
Decision Tree	max_depth	{5, 8, 10, 15, 20, 25, 30}	10	0.468	-
Random Forest	max_depth	{5,10,15,20,25,30, 50}	30	0.489	-
	n_estimators	{10, 20, 30, 50}	50		
MLP	hidden_layer	{[30], [15,15], [10,10,10]}	[15,15]	0.515	0.6956
	alpha	{0.01, 0.1, 1}	0.01		
AdaBoost	n_estimators	{20,40,60,100,200}	60	0.464	0.7236
	learning_rate	{0.01, 0.1, 1}	0.1		
Logistic Regression	C	{0.01, 0.1, 1, 10]	0.01	0.424	0.6743

We used PySpark to extract the descriptive statistics of the raw dataset, which are summarized in Table 2.

4.2 Intra-Subject Results

As the number of epochs contained in the SHHS dataset is very large (an epoch has a raw feature vector containing 7500 features. It needs more than 253 GB memory to store all the raw feature vectors), the training of deep learning models is extremely time-consuming. In our intra-subject experiments, we only considered the EEG data from the first 100 subjects in SHHS1. There are 98,985 epochs in total from the 100 subjects, in which we randomly selected 79,188 epochs (80%) as the training set and the rest 19,797 (20%) as the validation set.

4.2.1 Results of Basic Machine Learning Algorithms

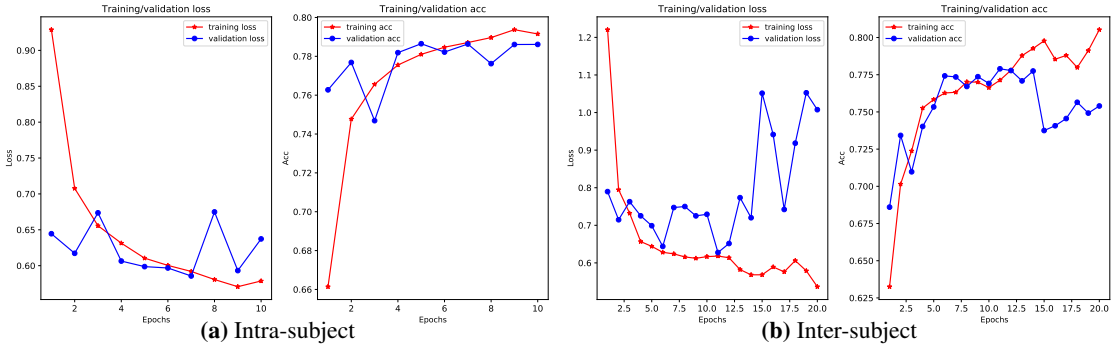
We first tried basic machine learning algorithms based on `scikit-learn` library[13], an open source machine learning library in Python. To tune the hyperparameters of the machine learning models, we applied `GridSearchCV` functionality provided by `scikit-learn` to conduct grid search based on cross validation where 5-fold cross validation was used. Table 3 shows the performance of basic machine learning algorithms. We studied in total 5 algorithms: decision tree, random forest, MLP, AdaBoost, and logistic regression. Some hyperparameters were tuned through `GridSearchCV`, while the rest of the hyperparameters were set to their default values defined in `scikit-learn`. As we can see, the basic machine learning models have much worse performance compared with the reports from Biswal et al.[3]. This may be due to the fact that we used a much smaller dataset. Furthermore, Biswal et al.[3] used 6 EEG channels but our data only have two.

4.2.2 Results of Deep Learning Algorithms

We trained three deep learning algorithms: CNN, RNN, and RCNN to do sleep stage classification. The implementations of these three deep learning algorithms are based on the high-level neural networks API `Keras` running on top of `Tensorflow`[13]. For all these three algorithms, batch size was set to be 128 and number of epochs was 10. Moreover, the lookback steps of RNN and RCNN were set to be 10. The architectures of these three neural nets are summarized as below:

1. CNN: 3 convolution layers each followed by a max-pooling layer with 1×2 pooling window. The three convo-

Figure 1: Loss and accuracy with epochs for CNN model (a) Intra-subject (b) Inter-subject



(a) Intra-subject **(b)** Inter-subject

Table 4: Performance of Deep Learning Algorithms

	CNN	RNN	RCNN
Our accuracy (intra-subject)	0.7863	0.8664	0.8388
Our accuracy (inter-subject)	0.7811	0.6928	0.6521
Biswal et al. accuracy	0.7731	0.7946	0.7981

lution layers have 64, 128, and 256 1×3 filters respectively. The output of the last pooling layer is connected to two fully connected layers. The first layer has 64 hidden units and has a dropout probability of 0.5. The last layer has 5 hidden units.

2. RNN: 5 layers of LSTM cells with dropout probability of 0.1, and each layer has 1000 units.

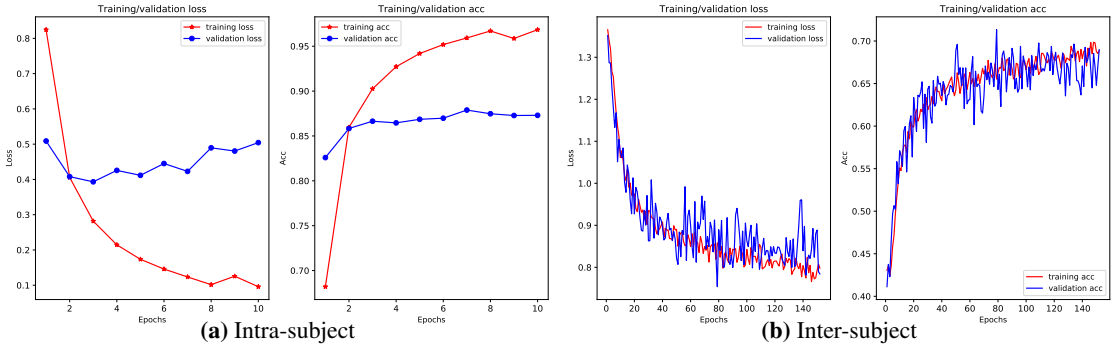
3. RCNN: 2 convolution layers and 4 LSTM layers. The first convolution layer has 32 1×3 filters which is followed by a max-pooling layer with 1×2 pooling window. The second convolution layer has 64 1×3 filters. All LSTM layers contain 1000 units, where each cell has a dropout probability of 0.1.

To find the best performed model without overfitting, we tracked the training loss and validation loss of each epoch which can are shown in Figure1, 2 and 3. The best accuracy was recorded when validation loss does not decrease further for each algorithm, which ensures that the trained model is free from overfitting. Specifically, CNN achieves best performance at epoch 7, RNN at epoch 3, and RCNN at epoch 7. The performances of our best found trained deep learning models are summarized in Table 4. Compared with basic machine learning approaches, deep learning algorithms achieve much higher accuracies. This demonstrates the advantages of deep learning in complex pattern learning. Furthermore, all of our trained deep learning models outperform those reported by Biswal et al.[3]. The reason, we believe, is that we used only intra-subject features during training. We will next show our inter-subject results.

4.3 Inter-Subject Results

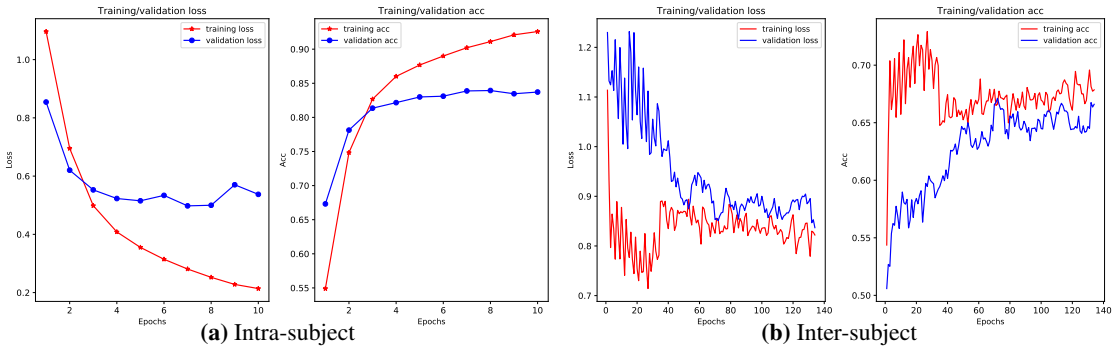
In inter-subject experiments, we used the first 5000 subjects for training, the next 200 subjects for validation, and the rest subjects for testing. Because every subject contains several thousands of epochs and each epoch contains 7500 features, it's impossible to load the entire training/validation/testing dataset into the memory at once. Therefore, we used data generator which generates batches of samples in real time and feeds them to the deep learning model. Specifically, we used `fit_generator` and `evaluate_generator` provided by `Keras`. We did not apply basic machine learning algorithms in inter-subject experiments because of their poor performance in intra-subject cases. Instead, we only focused on deep learning approaches and also tried ensemble methods to improve overall performance.

Figure 2: Loss and accuracy with epochs for RNN model (a) Intra-subject (b) Inter-subject



(a) Intra-subject **(b)** Inter-subject

Figure 3: Loss and accuracy with epochs for RCNN model (a) Intra-subject (b) Inter-subject



(a) Intra-subject **(b)** Inter-subject

4.3.1 Results of Deep Learning Algorithms

We used exactly same neural network architectures and hyperparameters for CNN, RNN, and RCNN as shown in Section 4.3.2. We also tracked the training loss and validation loss of each epoch to diagnose overfitting. In the case of CNN, the model starts to overfit after epoch 11. Therefore, we used the model trained at epoch 11 for testing. As shown in Table 4, CNN achieved an accuracy of 78.11% which is a little higher than Biswal et al's results. This is intriguing as we only have 5000 subjects for training with two channels of EEG signals for each subject. In contrast, Biswal et al surveyed 9000 subjects each with six channels. It is noticed, however, for RNN and RCNN, our results are much worse than theirs. From Figure 2b and 3b, we can see that the training loss and validation loss fluctuate widely. In general, we had expected to see that training loss decreases with increasing number of epochs. Also the validation loss should decrease until overfitting happens beyond which point it begins to increase. In the case of RNN and RCNN, we can see the validation loss decreases up to a certain level, then fluctuates widely. No "elbow" is observed for both algorithms. The wide fluctuation may be caused by the following reasons:

1. "Epoch" used in this project is different from the one commonly found in textbooks where one epoch refers to one pass through the entire dataset. In our case it means we should train all EEG signals from the first 5000 subjects in one epoch. We attempted this at beginning but soon realized this was extremely time-consuming and we could not obtain any result within two days. Therefore, in our current implementation, we set one epoch as 500 batches of samples. This setting guarantees that we can get some results, but it requires feeding the model with different data at different epochs. Since the model sees different data at different epochs, the training loss and validation loss fluctuated.

2. Learning rate decays inappropriately. Training/validation loss fluctuations are usually caused by a large learning rate. In our implementations, we used Adam optimizer which requires setting the initial learning rate and learning rate decay factor. We have experimented different combinations of both values, but failed to find

Table 5: Performance of ensemble methods

	unweighted average	majority voting	super learner (L2)	super learner (cross entropy)
accuracy	77.80	75.09	79.86	80.03

optimum combinations. With all paired values experimented, we observed consistently wide fluctuations that are similar to what are shown in Figure 2b and 3b.

Due to wide fluctuations, we cannot find the "elbow" which achieves lowest validation loss. Therefore, our results are much worse than Biswal's. Moreover, because of limitation of time and computation resource, it is not possible for us to do computation-intensive hyperparameter tuning. Overall our inter-subject results demonstrate that the neural network architectures suitable for intra-subject stage annotation seem not good for inter-subject classification task (except for CNN).

4.3.2 Results of Ensemble Approach

Since it was impractical to do computation-intensive hyperparameter tuning on deep neural networks in the short time-frame given, we instead tried ensemble approaches for performance improvement. We investigated three ensemble approaches: unweighted average, majority voting, and super learner. In unweighted average approach, we averaged the predicted probability after softmax transformation and did the classification based on the averaged probability. Although this approach is straightforward and simple, it is limited by the weaker learners and sensitive to the over-confident candidate[14]. Majority voting does not average the predicated probability but counts the votes of all labels from the base learners. The final prediction uses the label with most votes. This approach is less sensitive to the output from one single base learner, compared with unweighted average approach. However, when there are multiple dependent base learners, the predicated label will still be dominated. Super learner[15] uses cross validation results to optimize the weight vector in stacking base learners. Suppose we have M base learners and use K-fold cross validation. Then the cross validated loss is

$$R(\mathbf{a}) = \sum_{k=1}^{K} \sum_{i \in val(k)} \sum_{c=1}^{C} l(y_i^c, f(\mathbf{a}, \tilde{\mathbf{y}}_i^c)) \tag{1}$$

where $val(k)$ is the set of indices of the samples in the k-th fold, C is the number of classes, $\mathbf{a} = (a_1, a_2, ..., a_M)$ is the weight vector, $\tilde{\mathbf{y}}_i^c = (y_{i,1}^c, y_{i,2}^c, ..., y_{i,M}^c)$ is the prediction vector for i-th sample on class c, $f(\cdot)$ is the function to combine the predictions of all base learners, and $l(\cdot)$ is the loss function. Super learner will search the $\mathbf{a}$ which can minimize R and uses the optimized $\mathbf{a}$ to combine the predictions of all base learners for the final predictions on testing data. In this project, we use two loss functions: L2 loss and cross entropy loss. For L2 loss, $f(\mathbf{a}, \tilde{\mathbf{y}}_i^c) = \sum_{m=1}^{M} a_m y_{i,m}^c$. For cross entropy loss, $f(\mathbf{a}, \tilde{\mathbf{y}}_i^c) = softmax(\sum_{m=1}^{M} a_m logit(y_{i,m}^c))$. Furthermore, we only use 1-fold cross validation because of the long training time for deep learning models. Namely, we only validate the models on the validation set (i.e., 200 subjects) and thus optimize $\mathbf{a}$. With the optimized $\mathbf{a}$, we apply $f(\cdot)$ for ensembling prediction on the testing set.

The performance of these ensemble methods are shown in Table 5. As we can see, majority voting achieves a worse accuracy than the base learner CNN while unweighted average achieves a slightly worse performance than CNN. This is because we only get three base learners which are not diverse enough. In the case of super learners, both achieve a superior performance than the three base learners. It is found that the supper learner with cross entropy loss achieves the best performance (80%). Note that we achieved this best performance without any painful and computation-intensive hyperparameter tuning.

5 Conclusion

In this project, we used both basic machine learning algorithms and deep learning models (CNN, RNN, and RCNN) to automatically annotate sleep stages based on collected EEG signals. We fed raw features, as sampled from EEG signals, directly into various models, without carrying out feature engineering. We investigated two application scenar-

ios: intra-subject and inter-subject. In the intra-subject case, even when as few as 100 subjects were used, a stunning classification accuracy of 86.64% was reached. In addition, the intra-subject results demonstrate the advantages of deep learning algorithms compared with basic/shallow machine learning algorithms. However, for the inter-subject scenario, our results are much worse than the ones reported by Biswal et al.[3] except for CNN. This is possibly because we haven't done highly computation-intensive hyperparameter tuning due to limited time and computation resources. Instead, we tried ensemble approaches for performance improvement through linearly combining multiple trained deep learning models. Our results show that super learner method achieves the best performance among all investigated ensemble methods and outperforms the results Biswal et al.[3] without performing any computation-intensive hyperparameter tuning on deep neural nets.

References

1. David R Hillman, Anita S Murphy, and Pezzullo Lynne. The economic cost of sleep disorders. *Sleep*, 29(3):299–305, 2006.
2. C Iber, S Ancoli-Israel, A Chesson, and S. F Quan. The aasm manual for the scoring of sleep and associated events: rules, terminology and technical specifications. 2007.
3. Siddharth Biswal, Joshua Kulas, Haoqi Sun, Balaji Goparaju, M Brandon Westover, Matt T Bianchi, and Jimeng Sun. Sleepnet: Automated sleep staging system via deep learning. *arXiv preprint arXiv:1707.08262*, 2017.
4. O Tsinalis, Paul M Matthews, Y Guo, and S Zafeiriou. Automatic sleep stage scoring with single-chanel eeg using convolutional neural networks. *arXiv preprint arXiv:1610.01683v1*, 2016.
5. A Supratak, H Dong, C Wu, and Y Guo. Deepsleepnet: a model for automatic sleep stage scoring based on raw single-channel eeg. *arXiv preprint arXiv:1703.04046v2*, 2017.
6. H Dong, A Supratak, W Pan, C Wu, Paul M Mattews, and Y Guo. Mixed neural network approach for temporal sleep stage classification. *arXiv preprint arXiv:1610.06421*, 2016.
7. Dennis A Dean, Ary L Goldberger, Remo Mueller, Matthew Kim, Michael Rueschman, Daniel Mobley, Satya S Sahoo, Catherine P Jayapandian, Licong Cui, Michael G Morrical, et al. Scaling up scientific discovery in sleep medicine: the national sleep research resource. *Sleep*, 39(5):1151–1164, 2016.
8. Stuart F Quan, Barbara V Howard, Conrad Iber, James P Kiley, F Javier Nieto, George T O'connor, David M Rapoport, Susan Redline, John Robbins, Jonathan M Samet, et al. The sleep heart health study: design, rationale, and methods. *Sleep*, 20(12):1077–1085, 1997.
9. C O'Reilly et al. Montreal archive of sleep studies: an open-access resource for instrument benchmarking and exploratory research. *J. of Sleep Research*, 23(6):628–635, 2016.
10. A. L. Goldberger et al. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, pages 591–596, 2000.
11. B Kemp et al. Analysis of a sleep-dependent neuronal feedback loop: The slow-wave microcontinuity of the eeg. *EEE Trans. Biomed. Eng.*, 47(9):11851194, 2000.
12. T. K. Ho. Random decision forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pages 278–282, 1995.
13. Scikit-learn, keras and tensorflow. *http://scikit-learn.org/ and https://keras.io/ and https://www.tensorflow.org/*.
14. Cheng Ju, Aurélien Bibaut, and Mark van der Laan. The relative performance of ensemble methods with deep convolutional neural networks for image classification. *Journal of Applied Statistics*, pages 1–19, 2018.
15. Mark J Van der Laan, Eric C Polley, and Alan E Hubbard. Super learner. *Statistical applications in genetics and molecular biology*, 6(1), 2007.