School of Information Technologies
Faculty of Engineering & IT

ASSIGNMENT/PROJECT COVERSHEET - INDIVIDUAL ASSESSMENT

Unit of Study:	COMP5703 Information Technology
Assignment name:	Project Report
Tutorial time:	
Tutor name:	

DECLARATION

Student ID:	470144491
Student name:	Shaowei Zhang
Signed	张少为
Date	Nov 12, 2017

SIT Building, J12
The University of Sydney
NSW 2006 Australia

T +61 2 9351 3423
F +61 2 9351 3838
E sit.info@sydney.edu.au
sydney.edu.au/it

ABN 15 211 513 464
CRICOS 00026A

Sleep Analysis Based on Deep Belief Networks

Name: Shaowei Zhang

Student ID: 470144491

Unikey: szha5691

Contents

Abstract

Classifying sleep stages using machine learning methods is very meaningful to study sleep activity. EEG signals can be used for classification. According to R&K criterion, sleep stages can be separated into stage 1, stage 2, stage 3, stage 4, stage R and stage W. After comparing the performance of Deep Belief Networks, the train and test data is split from all the epochs and the ratio between different sleep stages is just as the original data source. Then the performances of traditional machine learning methods and mixed machine learning methods have been compared with that of Deep Belief Networks. The result shows that traditional methods have a slightly better performance. To improve the performance of Deep Belief Networks, Logistic Regression is combined with it to classify certain sleep stages. The improvement successfully enhances the performance. In addition, Hidden Markov Model is tested to modify the prediction and the conclusion shows that Hidden Markov Model is not recommended for this project.

1. Literature Review

Sleep is an important physiological activity, whose disorder can cause problems in health. Sleep is separated into several stages according to the rule of Rechtchaffen and Kales (R&K). There are two general stages: non-rapid eye movement (NREM) and rapid eye movement (REM). NREM can be further separated into stage 1 (S1), stage 2 (S2), stage 3 (S3) and stage 4 (S4). Traditionally, sleep stages are scored manually which is very time-consuming.

With the help of electroencephalogram (EEG) signals and machine learning, sleep stages can be scored automatically. Many methods have been proposed and most of them can be narrowed down into two types: traditional ways and deep learning ways. Sun and Zhang[1] classified time-varying electroencephalographic (EEG) signals from the aspect of updating feature extractors and proposed an adaptive feature extractor. They performed a support vector machine (SVM) classifier for multi-class mental imagery tasks and it shows the effectiveness of the proposed adaptive feature extractor. Güneş[2] used a novel data preprocessing method called k-means clustering based feature weighting (KMCFW). It combined with k-NN (k-nearest neighbor) and decision tree classifiers to classify the sleepstages. Li and Cui[3] proposed a hybrid automatic sleep stage scoring approach, named HyCLASSS. It uses random forest to classify the sleep stages. These methods firstly extract time-domain, frequency-domain or time-frequency-domain from recording epochs. Then the features are used to classify the sleep stages with traditional machine learning methods. Another kind of method is deep learning method. Supratak and Dong[4] proposed a deep learning model, named DeepSleepNet, for automatic sleep stage scoring based on raw single-channel EEG. It uses convolutional neural network to extract features to perform the classification. Li and Li[5] proposed a novel Deep Belief Networks (DBN) based model for affective state recognition from EEG signals. Signals from each EEG channel are firstly processed with a DBN for effectively extracting critical information from the over thousands of features. Then a supervised Restricted Boltzmann Machine (RBM) is applied on the combined low dimensional characteristics from the optimal EEG channels. Xu and Plataniotis[6] applied two types of semi-supervised deep learning approaches, stacked denoising autoencoder (SDAE) and deep belief networks (DBN) as application specific feature extractors for the affective states classification problem using EEG signals. The DBN based model achieved averaged F1 scores of 86.67%, 86.60% and 86.69% for arousal, valence and liking states classification respectively. Deep learning methods utilizes multiple layers of processing units to learn features from input data and then sleep stages are classified using supervised ways.

2. Methodology

The main method of this project is built based on a method proposed by Xia and Li [7].

The electroencephalogram (EEG) signals are used to perform automatic sleep scoring. In this study, deep belief network (DBN) is employed to discriminate sleep stages.

2.1 Deep Belief Networks (DBN)

The advantage of DBN is that it is suitable for data with only limited labels, which means it's very suitable for time series data like wave signals or music waves.

2.1.1 RBM

DBN is formed by stacking multiple layers of RBM. A restricted Boltzmann machine (RBM) is a generative stochastic artificial neural network that can learn a probability distribution over its set of inputs. The structure of RBM is showed in Fig. 2.1.

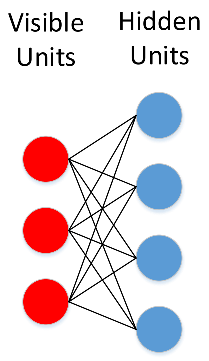


Fig. 2.1 restricted Boltzmann machine (RBM)

There're two layers in RBM, which are visible layer (visible units) and hidden layer (hidden units). Visible units can also be considered as input layer and hidden units can be considered as output layer.

First of all, the input data will be trained forward. Then the output of hidden units will be reconstructed backforward and the reconstructed result will be compared with the original input data. The output will only converge when the error between reconstruction and origin input data is acceptable.

2.1.2 Two process of DBN

Stack multiple layers of RBM then DBN can be constructed. There're two main processes in DBN, which are Pre-training and Fine-tuning.

2.1.2.1 Pre-training

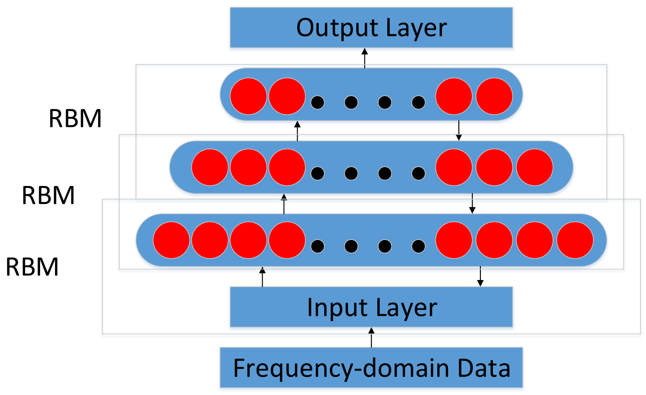


Fig. 2.2 Implement Process (Pre-training)

Fig. 2.2 is the implement process of "Pre-training". In a DBN, the output layer of the previous RBM is the input layer of the later RBM. The network is trained with RBM layer by layer. "Pre-training" is the unsupervised part of DBN.

Besides, as an unsupervised part of DBN, "Pre-training" actually can extract the features of the input data. The extracted features can also be applied in traditional methods apart from deep learning method.

2.1.2.2 Fine tuning

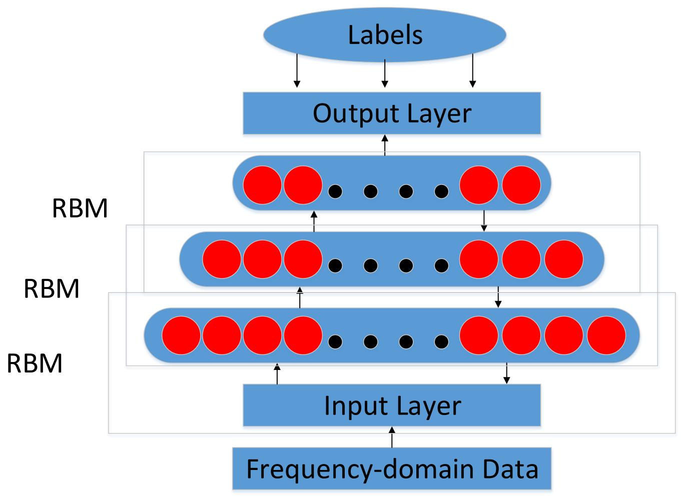


Fig. 2.3 Implement Process (Fine tuning)

After the network is well trained, true labels will be needed to perform a supervised training. Fig. 2.3 is the implement process of “Fine tuning”. With true labels, the resulting deep network will be tuned with gradient descent and backpropagation, which may lead to a better prediction. The labels will be reconstructed backward and be compared with the original input data. This process will be finished until the error between reconstruction and input data is acceptable.

2.2 Data Preprocessing

2.2.1 Dataset

The dataset used in this study is obtained from the SC* files in sleep-EDF Database [Expanded][8-9]. 20 healthy experiment subjects (which will be referred as participants), aged from 25 to 33, are included in this dataset. Two PSGs of about 20 hours each were recorded during two subsequent day-night periods at the subjects' homes. The second night of the 14th subject was lost and total 39 PSGs were used. The experiment subjects included in the dataset are 10 males and 10 females, and were 25-34 years old at the time of the recordings. Each EEG signal was sampled at 100 Hz. There’re two channels in the SC* files. After comparison, EEG Fpz-Cz channel is more characteristic. Fig. 2.4 shows one example of the EEG Fpz-Cz signals.

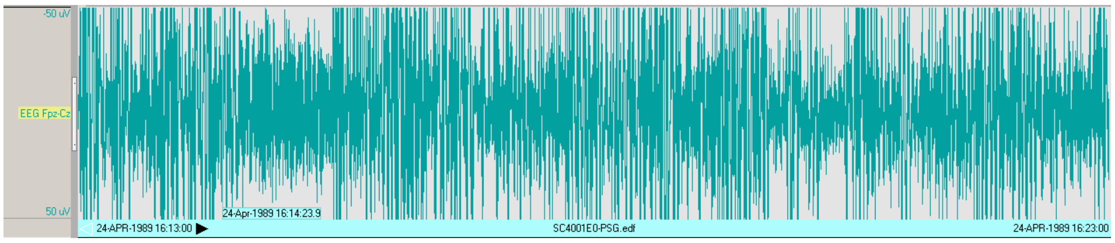


Fig. 2.4 EEG Fpz-Cz channel

The corresponding labels come from the *Hypnogram.edf files in sleep-EDF Database [Expanded]. The *Hypnogram.edf files contain annotations of the sleep patterns that correspond to the PSGs. These patterns (hypnograms) consist of sleep stages W, R, 1, 2, 3, 4, M and ?. Stage W means the stage person is awake. Stage R means the stage person has rapid eye movement. Stage 1,2,3,4 means different sleep stages. The larger the index is, the deeper the sleep is. Stage M means the stage in movement time.

Stage ? means the stage has not been scored. All hypnograms were manually scored by well-trained technicians every 30 seconds.

2.2.2 Sampling

According to the R&K criterion [10], sleep states consist of two general stages: non-rapid eye movement (NREM) and rapid eye movement (REM). NREM includes four stages: stage 1 (SI), stage 2 (S2), stage 3 (S3) and stage 4 (S4). To avoid that the mismarked epochs would affect the automatic scoring result, 'Stage M' and 'Stage ?' will be removed in this study because these two stages have not been referred in the R&K criterion. Samples are randomly picked from all the epochs, which is different from what Xia and Li [7] did. They chose the first 250 epochs in each night data. To avoid the centrality of sleep stages, this study applied a different method and randomly picked samples from all the epochs. This avoid the probability that samples from early time series may be only some shallow sleep stages.

2.2.2.1 Splitting train and test data

There're two ways to extract train and test data. The first way is to collect all the samples into one dataset and then split the dataset into training dataset and test dataset. In this way, epochs from all the participants are mixed and both training dataset and test dataset will include epochs from all the participants. The second way is to collect samples from part of the participants as a training dataset and then use samples from the rest of the participants as a test dataset. In this way, the samples in the test dataset will be totally new ones for the training dataset. After first thought, the second way is more practical than the first one. However, it may need the support of sufficient participant numbers.

2.2.2.2 Sample ratio selection

Another problem of sampling is how to deal with the ratio of the samples from different sleep stages. The ratio of samples for each sleep stage among all the epochs is showed in table 2.1. To simulate the true ratio between sleep stages, we need to follow this. However, the ratio indicates an obvious imbalance. The ratio of sleep stage

W is 68.02%, which is much higher than other stages. The ratio of sleep stage 1 and 4 are approximately 2%. Whether this imbalanced dataset is reasonable needs further research.

Table 2.1 Ratio of samples for each sleep stage

Sleep stage	W	1	2	3	4	R	All
Count	72354	2804	17799	3370	2333	7717	106377
Percentage (%)	68.02%	2.64%	16.73%	3.17%	2.19%	7.25%	100.00%

2.2.3 Fourier transformation

The Fourier transform decomposes time-domain signals into the frequencies that make it up [11]. The following couple of formulas are used to define it:

$$f(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx \qquad \text{(formula 2.1)}$$

$$f(x) = \int_{-\infty}^{\infty} f(\xi) e^{2\pi i \xi x} d\xi \qquad \text{(formula 2.2)}$$

Formula 2.1 transforms time-domain data into frequencies and formula 2.2 is used to transform frequencies back to time-domain data.

2.3 Logistic Regression (LR)

Logistic regression is a supervised classifier. Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variable that determine an outcome [12]. LR is a discriminative model for binary classification.

$$\mathrm{p}(y|x,w) = Ber\big(y|\sigma(\eta)\big) = \sigma(\eta)^{y}\big(1-\sigma(\eta)\big)^{1-y} \qquad \text{(formula 2.3)}$$

$$\eta = \mathrm{w}^{T} x \qquad \text{(formula 2.4)}$$

$$\sigma(\eta) \overset{\mathrm{def}}{=} \frac{1}{1+exp(-\eta)} = \frac{e^{\eta}}{e^{\eta}+1} \qquad \text{(formula 2.5)}$$

Logistic regression is the source of the sigmoid function used in backpropagation. Fig 2.5 shows the sigmoid function.

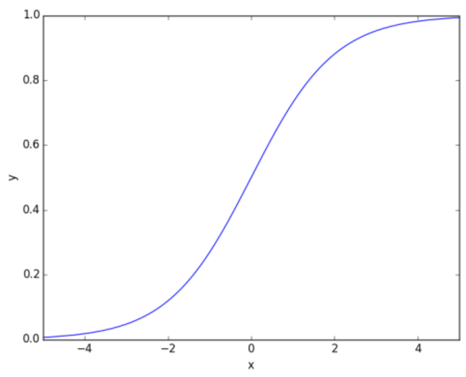
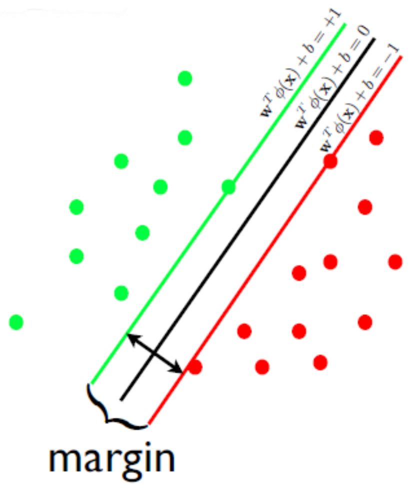


Fig. 2.5 Sigmoid function

It is equivalent to a one-layer backpropagation neural net. Logistic regression is basically a linear model and the weights are set during training to maximize the conditional data likelihood.

2.4 Support Vector Machine (SVM)

SVM is a supervised classifier. A support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space [13]. SVM is used to find the maximum margin between two classes.



$\mathbf{w}^T\phi(\mathbf{x})+b=+1$
$\mathbf{w}^T\phi(\mathbf{x})+b=0$
$\mathbf{w}^T\phi(\mathbf{x})+b=-1$
margin

Fig. 2.6 Margin between two classes

Considering soft margin, SVM will be a quadratic programming.

$$\arg\min C\sum_{n=1}^{N}\xi_n+\frac{1}{2}\|w\|^2 \qquad \text{(formula 2.6)}$$

$$\text{s.t.}\, t_n y(x_n)\geq 1-\xi_n, \xi_n\geq 0\ (n=1,2,\cdots,N)$$

2.5 Gaussian Naïve Bayes (GNB)

Gaussian Naïve Bayes is a supervised classifier. Bayes' Theorem is stated as [14]:

$$\mathrm{P}(h|d) = \frac{P(d|h)\cdot P(h)}{P(d)} \qquad \text{(formula 2.7)}$$

P(h|d) is the probability of hypothesis h given the data d. This is called the posterior probability. P(d|h) is the probability of data d given that the hypothesis h was true. P(h) is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h. P(d) is the probability of the data (regardless of the hypothesis).

Naïve Bayes means all the attributes are assumed to be conditionally independent. Gaussian Naïve Bayes means Gaussian function is used to estimate the distribution of the data.

2.6 Hidden Markov Model (HMM)

Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved states [15]. HMM consists two sequences. One is called states, which is the latent sequence. States is the sequence with maximum likelihood. Use {S1, S2, …, SN} to represent the states and { q1, q2, …, qN } to represent the sequence positions. Initial state distribution can be represented by $\pi_i = P[q_1 = S_i]$. State transition probabilities can be represented by $a_{ij} = P\left(q_{t+1} = S_i | q_t = S_j\right)$, which means the probability of moving to a given state depends only on the current state.

The other sequence is called observations. Observations is the sequence that truly observed. The probability of one observation can be defined as emission probabilities, which is $b_i(k) = P(o_t = k | q_t = S_i)$.

Here's an example of HMM. Suppose there're three kinds of weather, which are "Sunny", "Rainy", "Snowy". So the states are {Sunny, Rainy, Snowy}, Initial state distribution can be $\pi_i = (0.7\ 0.25\ 0.05)$. It means the prior probability of these states.

0.7 is for a sunny day. 0.25 is for a rainy day and 0.05 is for a snowy day. The state transition probabilities can be represented as A:

$$\mathrm{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} 0.80 & 0.15 & 0.05 \\ 0.38 & 0.60 & 0.02 \\ 0.75 & 0.05 & 0.20 \end{pmatrix}$$

In which for instance, $a_{12} = P\left(q_2 = S_{rainy} | q_1 = S_{sunny}\right)$. The probability of the sequence of "Sunny, Rainy, Snowy" will be 0.0021 as follow.

$$P_{sequence} = P\left(q_1 = S_{sunny}\right) \cdot P\left(q_2 = S_{rainy} | q_1 = S_{sunny}\right)$$

$$\cdot P\left(q_3 = S_{snowy} | q_2 = S_{rainy}\right) = 0.7 \times 0.15 \times 0.02 = 0.0021$$

Suppose there're three kinds of wearing style for people, which is {"T-shirt", "coat", "umbrella"}. These wearing style can be called observations. Their emission probability can be represented by B:

$$\mathrm{B} = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} 0.60 & 0.30 & 0.10 \\ 0.05 & 0.30 & 0.65 \\ 0.00 & 0.50 & 0.50 \end{pmatrix}$$

In which for instance, $b_{12} = P\left(o_2 = coat | q_2 = S_{rainy}\right)$. The probability of the observation sequence of "coat, coat, umbrella, umbrella, T-shirt, umbrella, umbrella" will be as follow.

$$P_{observation} = P(O_{coat}, O_{coat}, O_{umbrella}, O_{umbrella}, O_{T-shirt}, O_{umbrella}, O_{umbrella})$$
$$= \sum_{all\ Q} P(O|Q)P(Q)$$

The relationships between states and observations are abstractly showed in Fig 2.7.

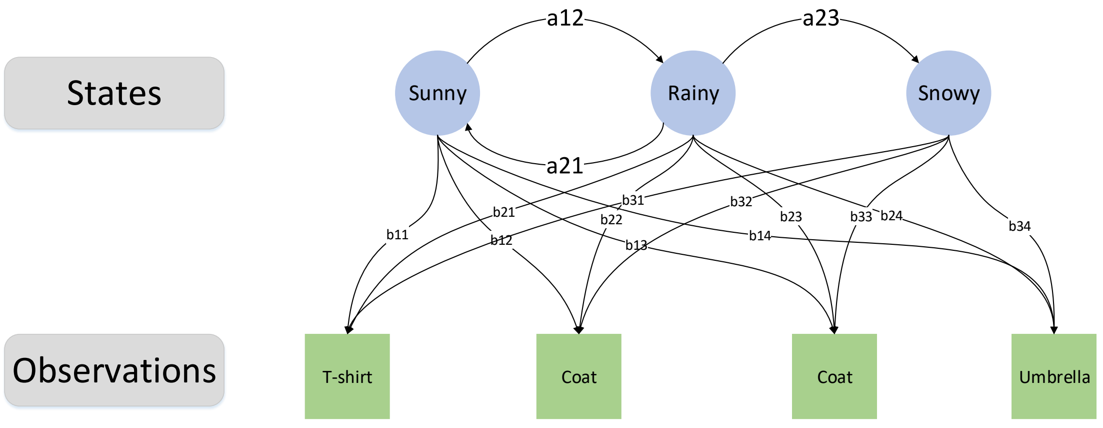


Fig. 2.7 Relationships between states and observations

The process of training a HMM is to find these probabilities with maximum likelihood. The prediction of DBN perhaps could be modified to get a better performance by applying a HMM.

3. Data processing

3.1 Fast Fourier transform (FFT)

Each sample is a 30s time series (which will be referred as one epoch) and this study converts them from time-domain to frequency domain using Fast Fourier transform (FFT). Fourier analysis converts a signal from its original domain like time to a representation in the frequency domain [11]. An FFT can rapidly perform the Fourier transformation. After FFT, the raw EEG signals are transformed into frequency-domain and only band between 0.5 Hz and 35 Hz will be picked. Fig. 3.1 shows one example of the FFT results. In order to extract stable features, the band is selected from 0.5 Hz to 35 Hz.

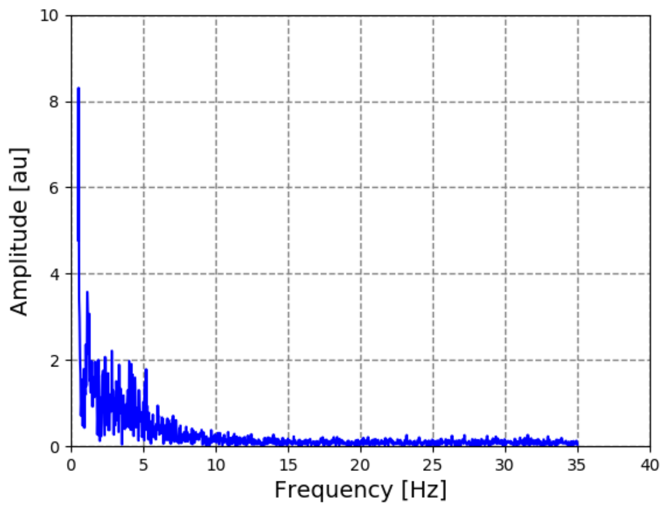


Fig. 3.1 One example of FFT results

After further scaling, the frequency-domain data will be used as the initial input data in the implement process.

3.2 Splitting train and test data

As mentioned in part 2, there’re two ways to split train and test data. The first way is to collect all the samples in one dataset and then split the dataset into training dataset and test dataset. The second way is to collect samples from part of the participants as a training dataset and then use samples from the rest of the participants as a test dataset. The latter way is more practical but considering our limited data source, the result of it may not be convincing.

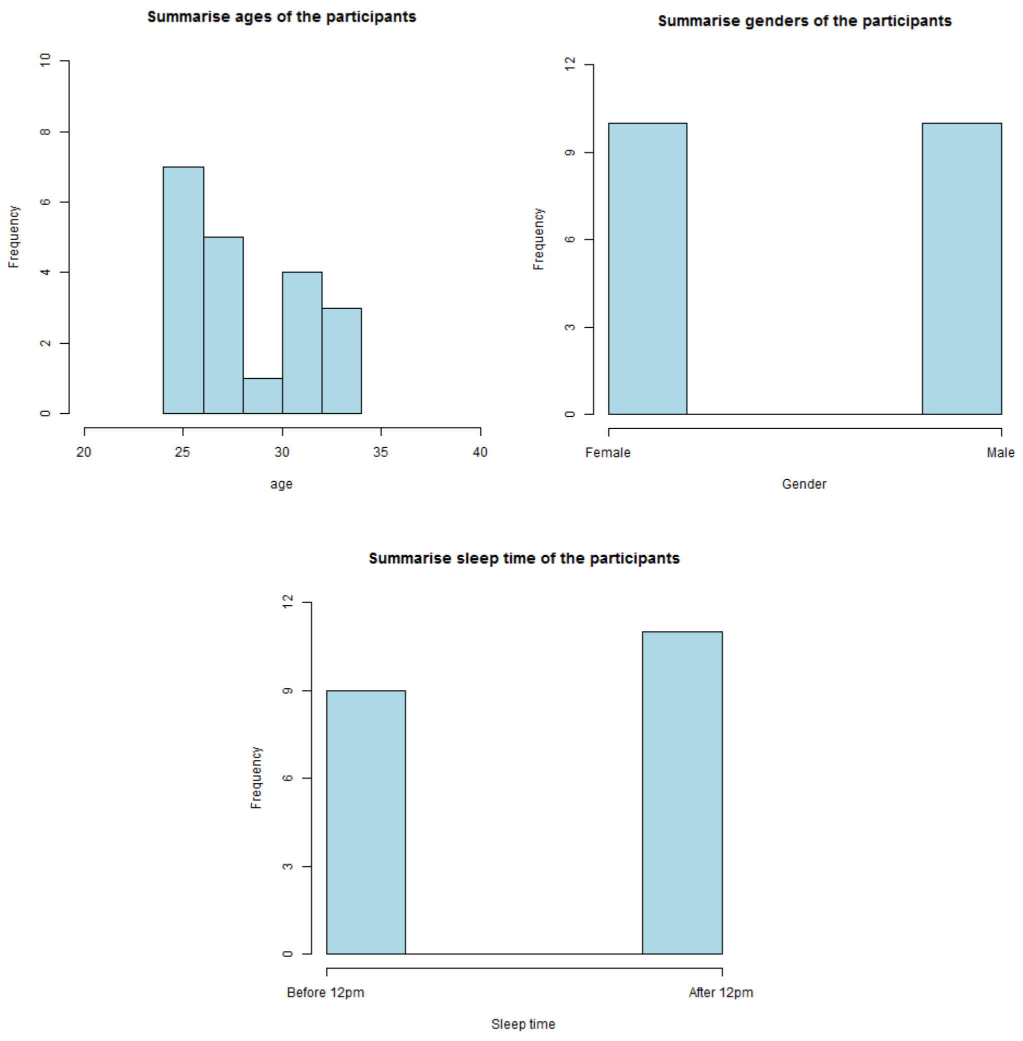


Fig. 3.2 Histograms of participants

Fig. 3.2 summarizes the frequencies of age, sleep time and gender of the participants. There’re only 20 participants but their ages range from 25 to 33, which means there’re only very few samples in one certain age range. Besides, half of them are male and half of them are female. Nearly half of them sleep before 12pm and nearly half of them sleep after 12pm. The EEG signals may differ a lot between different ages, gender and sleep time. For instance, young people may have a better sleep than the elderly. The male may have a different sleep from the female. People sleep after 12pm may have a worse sleep. In a word, each participant may be quite distinctive. The prediction maybe not very reliable with such a limited number of participants. These two ways should be compared to choose an appropriate one for further study.

3.3 Sample ratio selection

There're 6 sleep stages for classification. According to table 1, the ratio of samples for each sleep stage among all the epochs is quite imbalanced. Whether this ratio is reasonable for classification needs further research.

To use a comparatively reasonable ratio, the variance of the percentage is used as the measurement of the dataset's ratio. For instance, the variance of the percentage in table 1 is 0.055. If the sample numbers of different sleep stages are nearly the same, then the variance will be approximately 0.

As mentioned in part 3.2, two ways to split train and test data should both be considered. One is split the test data from all the epochs and the other way is to use part of the participants as test data. 7 different groups for each of ratio are arranged for each way and their variance range from 0.06 to 0 (from high variance to low variance). The detailed variances are showed in table 3.1.

Table 3.1 Variances for two way to split train and test data

Index	1	2	3	4	5	6	7
Way 1	0.00262	0.00727	0.01386	0.02674	0.03521	0.04435	0.05710
Way 2	0.00152	0.00567	0.01177	0.02423	0.03353	0.04371	0.05704

To perform a fair comparison, the ratios of way 1 and way 2 with the same index in table 2 are approximately similar.

4. DBN Implement

To get the best performance of the classifier, the best parameter should be chosen. The parameter range will be described in 4.1 and 4.2.

4.1 Pre-training

In the "deep-belief-network-master" package [16], there're some parameters for the "Pre-training".

- hidden_layers_structure: This is the number of the units in hidden layer. Referring to Xia and Li [7], the range of this parameter is designed from 100 to 500.
- batch_size: To reduce the memory occupation, training data will be divided into small batches and this is the size of batch. Referring to Xia and Li [7], the range of this parameter is designed from 10 to 100.
- learning_rate_rbm: This is learning rate of RBMs. The range of this parameter is designed from 0.01 to 0.1.
- n_epochs_rbm: This is the number of RBMs. The range of this parameter is designed from 10 to 20.
- activation_function: This is the activation function of units. There're two choices, "relu" and "sigmoid". The "relu" means rectifier, who is defined as $\mathrm{f}(x) = max(0, x)$. "relu" is showed in Fig. 4.1. The "sigmoid" is showed in Fig. 4.2.

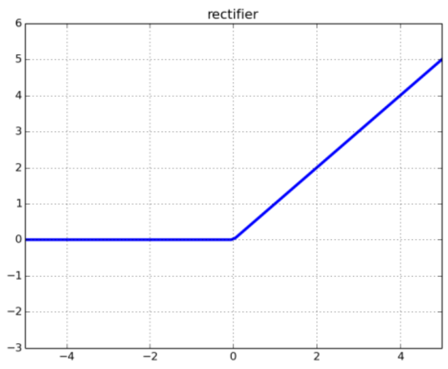


Fig. 4.1 "relu" function

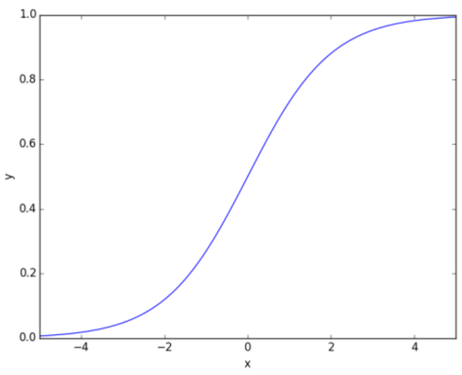


Fig. 4.2 "sigmoid" function

4.2 Fine-tuning

In the “deep-belief-network-master” package, apart from the parameters in “Pre-training”, there’re still some other additional parameters for “Fine tuning”.

- learning_rate: This is the learning rate of tuning. The range of this parameter is designed from 0.01 to 0.1.
- n_iter_backprop: This is limit number of backpropagation iteration. The range of this parameter is designed from 100 to 1000.
- dropout_p: This is the dropout percentage, which is used to avoid overfitting. The range of this parameter is designed from 0.1 to 0.2.

5. Sample selection

5.1 Comparison of metrics

With the parameters above, DBN will be used for the datasets in table2. The best parameter will be chosen and the metrics such as precision, recall and f1 score will be compared for each sleep stage and their average performance.

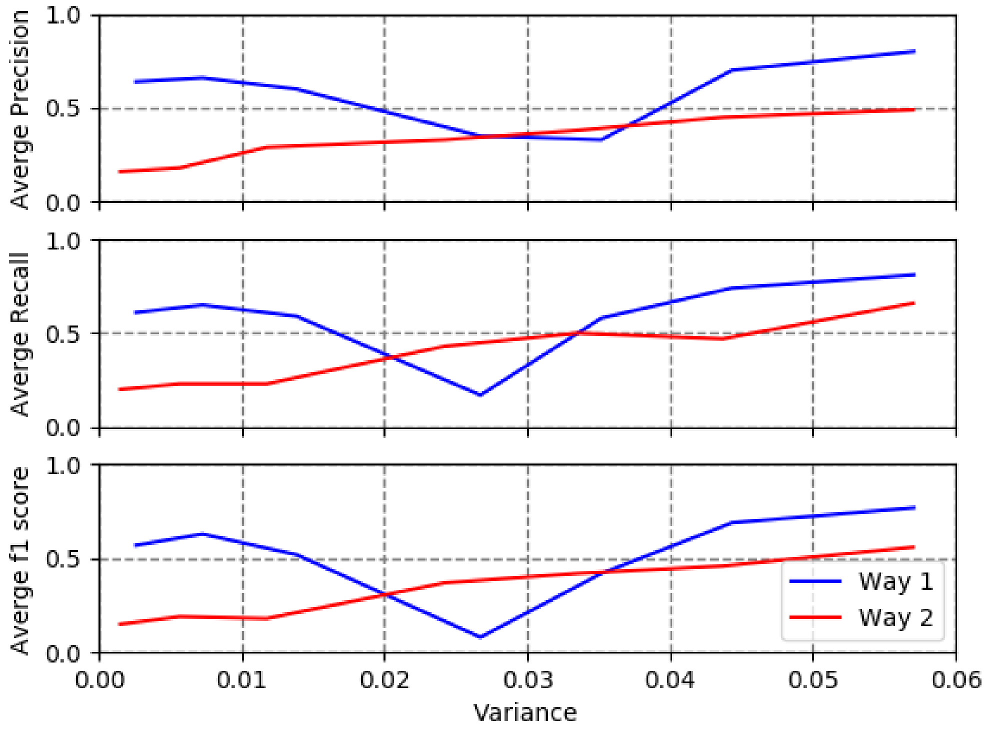


Fig. 5.1 Comparison for average metrics of all the sleep stages

Fig 5.1 shows the average metrics of all the sleep stages. It shows that way 1 has comparatively higher metrics than way 2. When the variance is 0 or 0.06, all the metrics of way 1 are good.

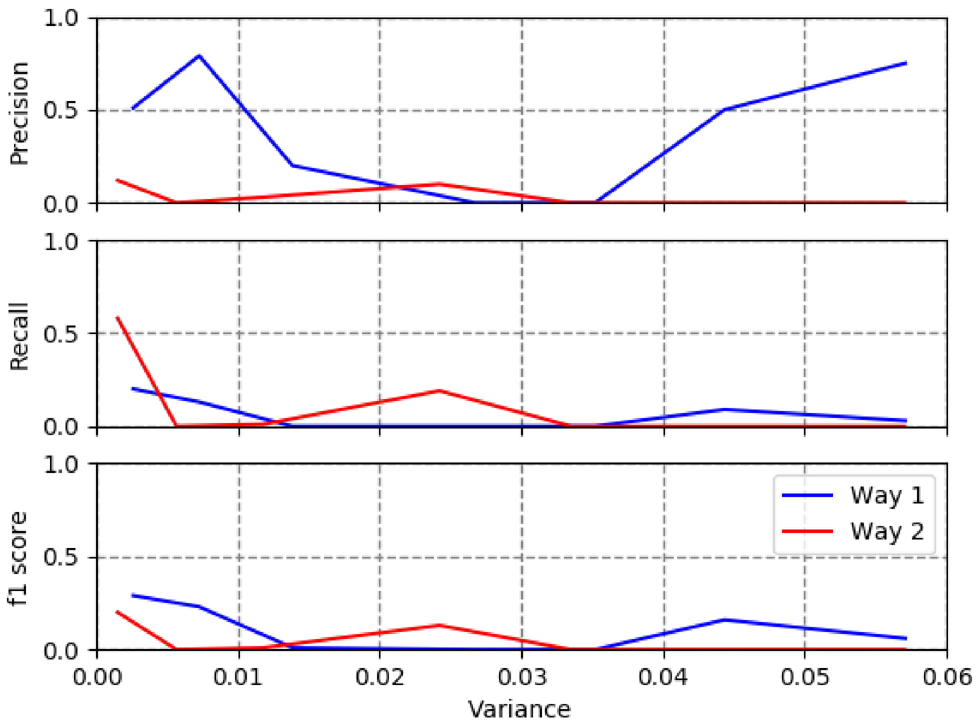


Fig. 5.2 Comparison for metrics of sleep stage 1

Fig 5.2 shows the metrics of sleep stage 1. The percentage of sleep stage 1 is 2.64%. It's a class with comparatively less samples. Fig 5.2 shows that for precision, way 1 is comparatively higher than way 2. Besides, when the variance is approximately 0.005 or 0.06, the precision is quite good. However, both recall and f1 score are not acceptable.

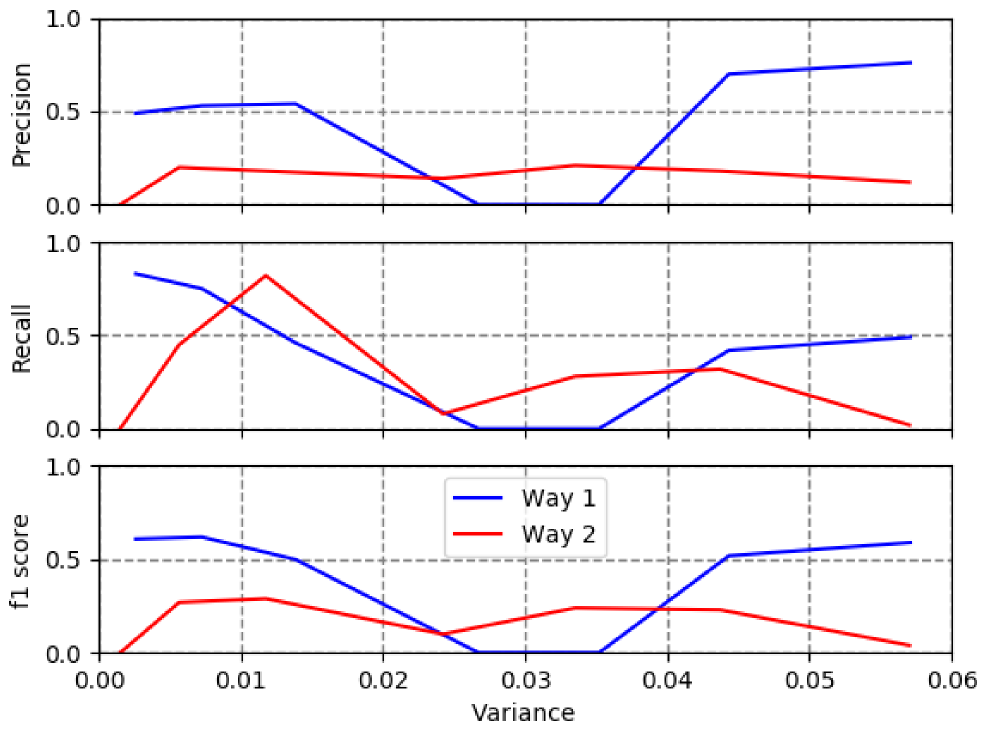


Fig. 5.3 Comparison for metrics of sleep stage 2

Fig 5.3 shows the metrics of sleep stage 2. The percentage of sleep stage 2 is 16.73%. It's a class with comparatively more samples. Fig 5.3 shows that for precision, way 1 is comparatively higher than way 2. When the variance is approximately 0.015 or 0.06, the precision is quite good. For recall, way 2 is comparatively higher than way 1, but when the variance is approximately 0, the recall of way 1 is better. For f1 score, way 1 is comparatively higher than way 2. When the variance is approximately 0 or 0.06, the precision is good.

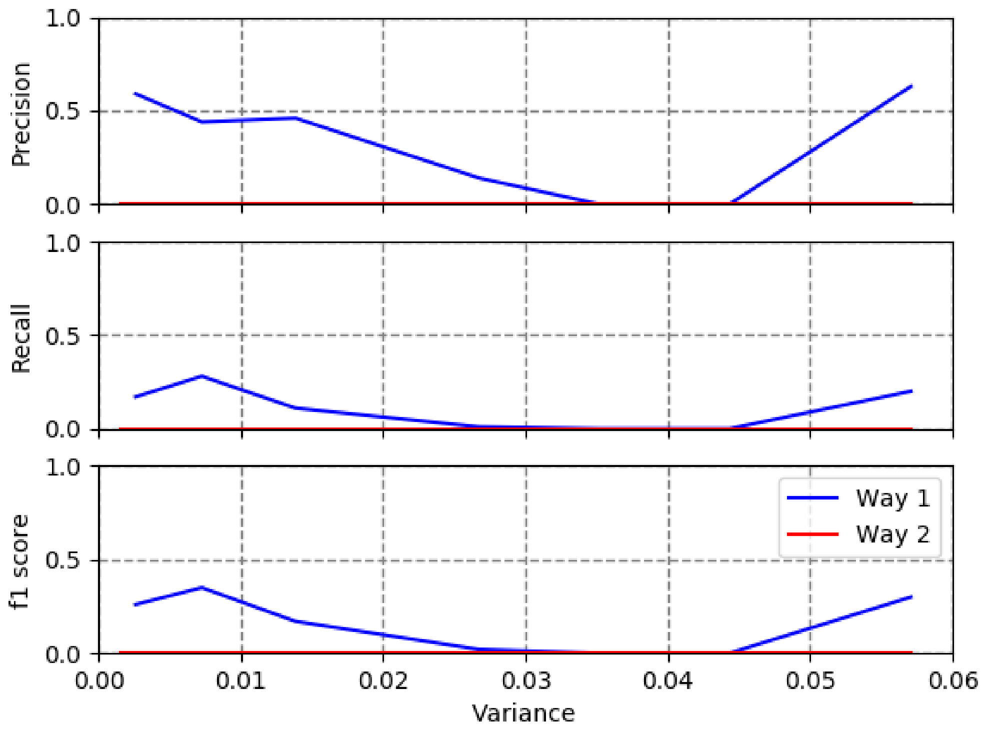


Fig. 5.4 Comparison for metrics of sleep stage 3

Fig 5.4 shows the metrics of sleep stage 3. The percentage of sleep stage 3 is 3.17%. It's a class with comparatively less samples. Fig 5.4 shows that for all the metrics, way 1 is comparatively higher than way 2. When the variance is approximately 0 or 0.06, the precision is good. However, all the metrics are too low, which are not acceptable.

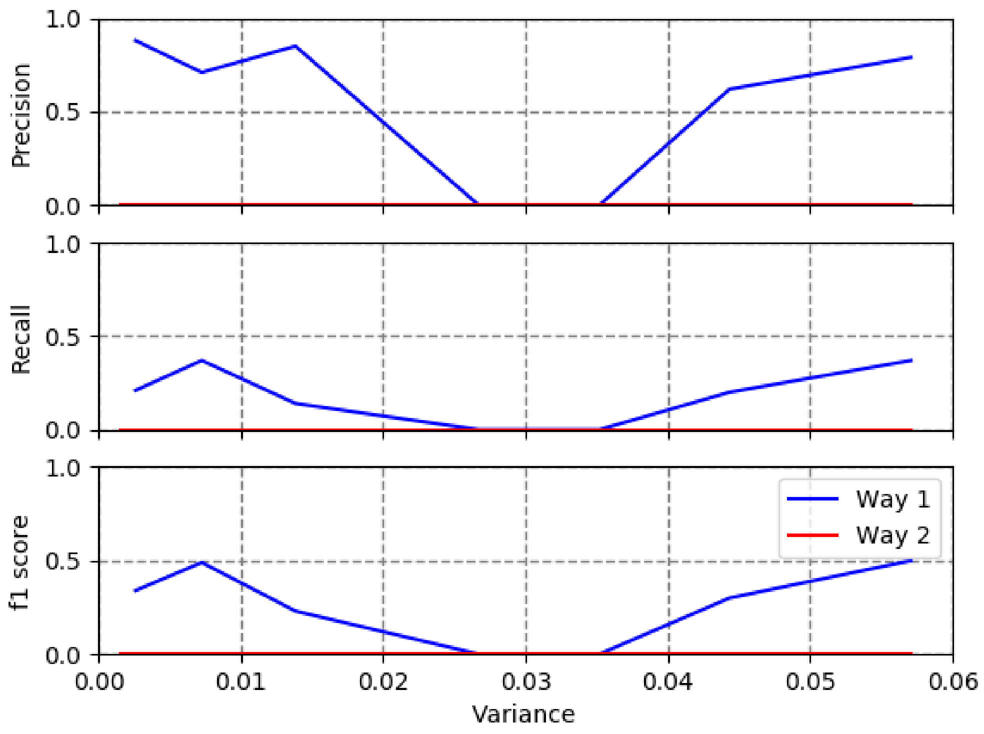


Fig. 5.5 Comparison for metrics of sleep stage 4

Fig 5.5 shows the metrics of sleep stage 4. The percentage of sleep stage 4 is 2.19%. It’s a class with comparatively less samples. Fig 5.5 shows that for precision, way 1 is comparatively higher than way 2. Besides, when the variance is approximately 0, 0.015 or 0.06, the precision is quite good. However, both recall and f1 score are not acceptable.

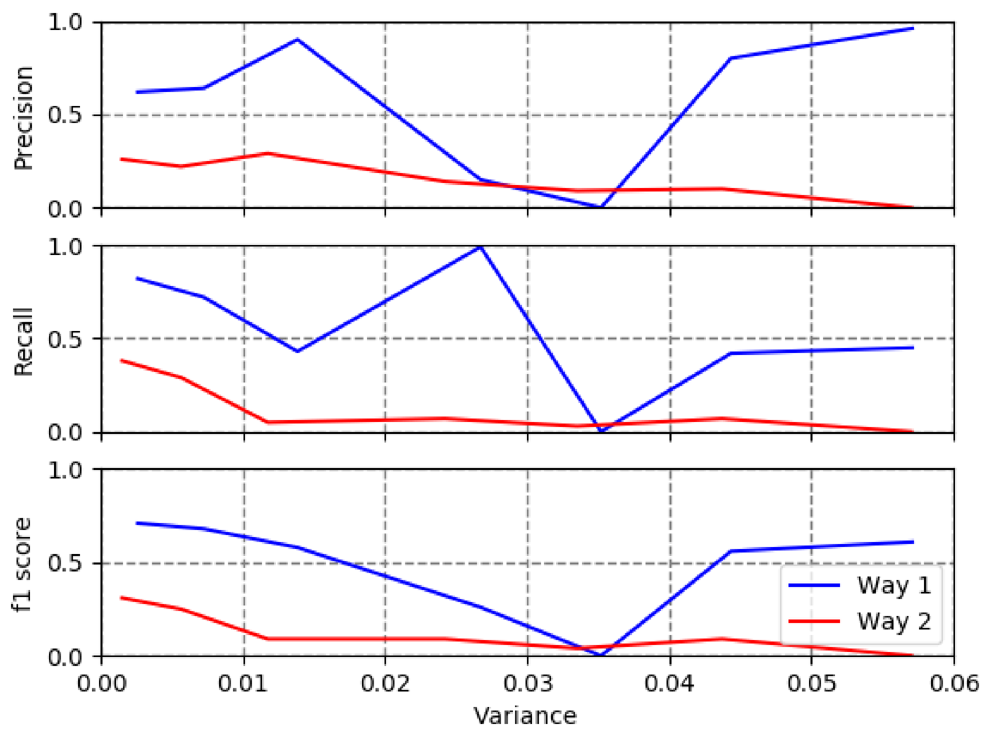


Fig. 5.6 Comparison for metrics of sleep stage R

Fig 5.6 shows the metrics of sleep stage R. The percentage of sleep stage R is 7.25%. It's a class with comparatively more samples. Fig 5.6 shows that for precision, way 1 is comparatively higher than way 2. When the variance is approximately 0.015 or 0.06, the precision is quite good. For recall, way 1 is comparatively higher than way 2. When the variance is approximately 0 or 0.025, the recall of way 1 is good. For f1 score, way 1 is comparatively higher than way 2. When the variance is approximately 0 or 0.06, the precision is good.

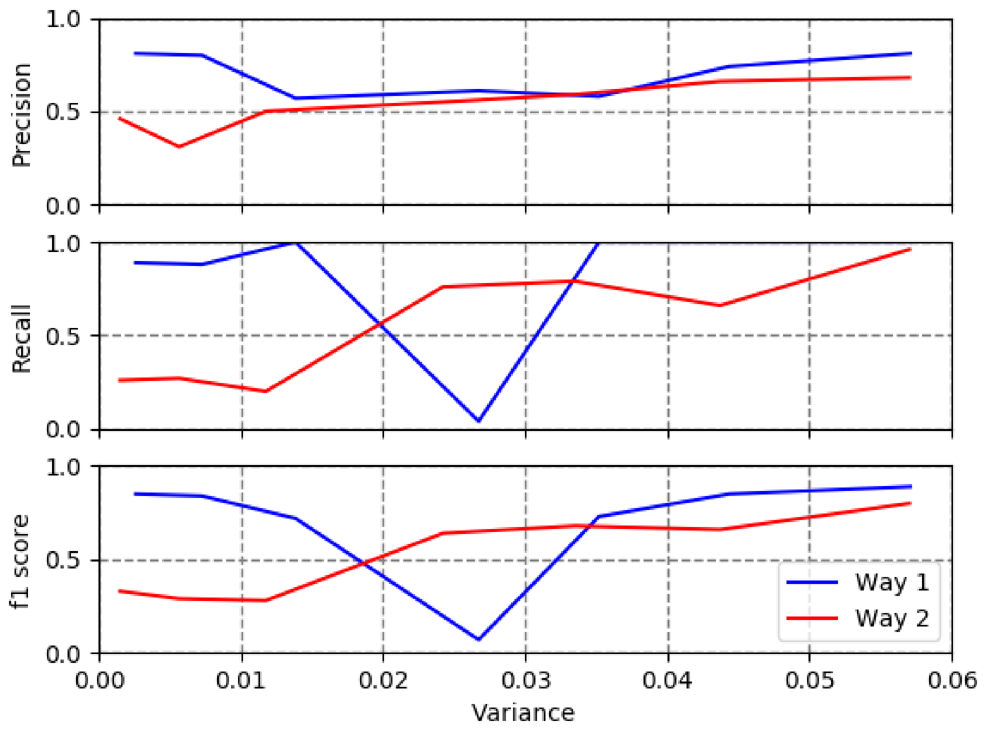


Fig. 5.7 Comparison for metrics of sleep stage W

Fig 5.7 shows the metrics of sleep stage W. The percentage of sleep stage R is 68.02%, which occupies most of the samples. Fig 5.7 shows that for precision, way 1 is comparatively higher than way 2. When the variance is approximately 0 or 0.06, the precision is quite good. For recall, way 1 is comparatively higher than way 2. Except when the variance is approximately 0.025, all the recall of way 1 is quite good. For f1 score, way 1 is comparatively higher than way 2. Except when the variance is approximately 0.025, all the f1 score of way 1 is quite good.

To sum up, way 1 has a comparatively better performance than way 2. Theoretically, way 2 is more practical but way 1 is more like the case that we have plenty of training data. If we choose way 2 for further study, the performance of our classifier may be underestimated because of our limited data source. With the application of machine learning methods for sleep stages, there will be more data source in the future so choosing way 1 for further study will be more meaningful for our project.

Besides, after comparing the metrics for each sleep stage, it could be found out that comparatively, when the variance is 0 or 0.06, the metrics are higher. To select the best variance for further study, we need a comparison.

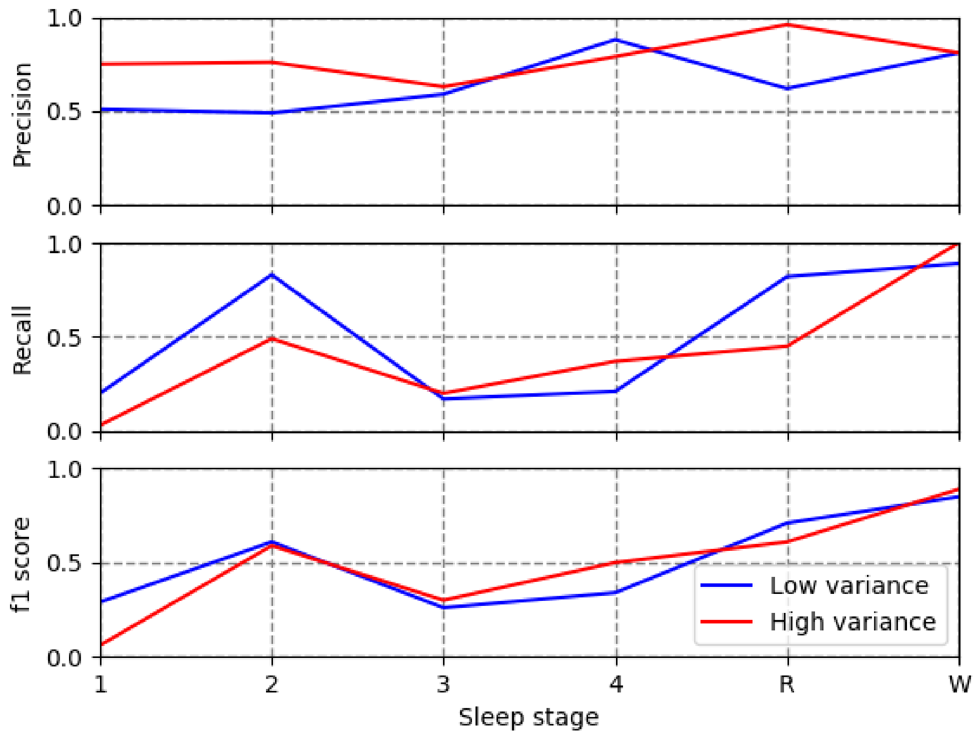


Fig. 5.8 Comparison between low variance samples and high variance samples

After comparing two kinds of samples, we find that there's no obvious advantage for any kind because higher precision means lower recall. The performance of low variance dataset is slightly steady while the performance of high variance dataset changes a lot for different sleep stages. Take recall as an example, the recall for sleep stage 1 is almost 0 but the recall for sleep stage W is 1. Considering our basic requirement, the performance of classifying sleep stage W should be as good as possible. The classification will not be acceptable if it mixes asleep state and awake state. Besides, samples with high variance is more practical. The sample of sleep stage W will indeed occupies most of the sample in reality just as our data source because the time people is awake is more than the time people is asleep. To conclude, we will choose high variance samples for further research. How to improve the performance of our classifier for other sleep stages will be discussed later.

6. Classifier selection

After the train and test dataset is confirmed, the performance of different classifiers should be compared with the performance of DBN. This study has performed three kinds of methods for classification. The first kind is deep learning method. DBN is applied. The second kind is traditional method. Logistic Regression, Support vector machine and Gaussian Naïve Bayes are used. The third kind is a combination of deep learning method and traditional method. In this part, RBM is used to extract features and then traditional methods will use these features to perform a classification. The combination of RBM and Logistic Regression is performed, which will be referred to as "RBM-Logistic". The purpose of this comparison can be narrowed down into 2 points. First, the performance of different classifiers for sleep stages could be known. Their performance could be compared with DBN.

Besides, this comparison is to see which method we should focus on in the further research. If some of these classifiers is quite poor, then it does not need to be considered in the further research at all.

The result below is based on the samples with a variance of 0.0571, which almost has the same ratio as the original dataset. 20% of the samples are separated as test data and the rest are training data. Use the best parameters for the models. The detailed prediction is showed in the following figures and tables.

6.1 Performance of different classifiers

Based on the same dataset, the performance of different classifiers are showed below. To compare with choosing dataset with high variance, the performance of classifiers using dataset with low variance will also be showed below.

6.1.1 DBN

The metrics of DBN classification are showed in table 6.1-1 and table 6.1-2.

Table 6.1-1 Classification result of DBN (low variance)

	Precision	Recall	f1-score	support
Sleep Stage 1	0.51	0.2	0.29	263
Sleep Stage 2	0.49	0.83	0.61	488
Sleep Stage 3	0.59	0.17	0.26	312
Sleep Stage 4	0.88	0.21	0.34	218
Sleep Stage R	0.62	0.82	0.71	489
Sleep Stage W	0.81	0.89	0.85	454
average/total	0.64	0.61	0.57	2224

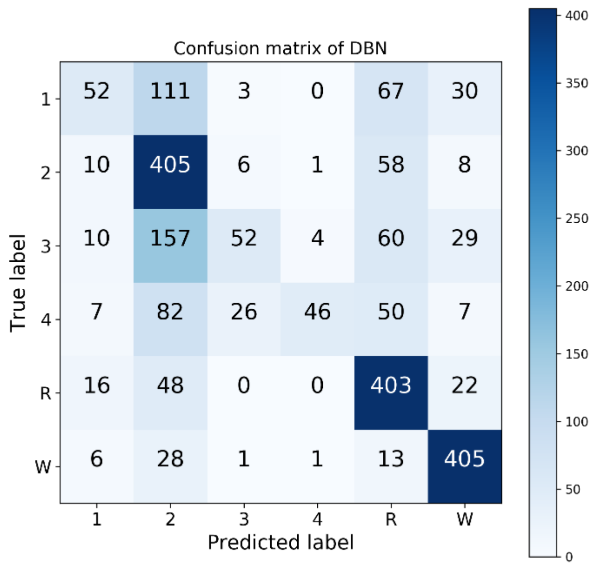
Table 6.1-1 shows that with low variance, DBN has a steady performance for each sleep stage. However, the recall for sleep stage 1, sleep stage 3 and sleep stage 4 is very low.

Table 6.1-2 Classification result of DBN (high variance)

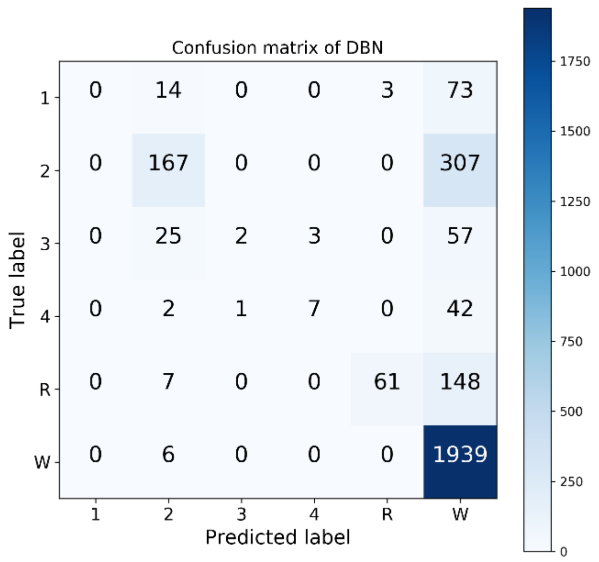
	Precision	Recall	f1-score	support
Sleep Stage 1	0.75	0.03	0.06	90
Sleep Stage 2	0.76	0.49	0.59	474
Sleep Stage 3	0.63	0.2	0.30	87
Sleep Stage 4	0.79	0.37	0.50	52
Sleep Stage R	0.96	0.45	0.61	216
Sleep Stage W	0.81	1.00	0.89	1945
average/total	0.80	0.81	0.77	2864

Table 6.1-2 shows that with high variance, DBN has comparatively high precision for each sleep stage. However, except for sleep stage W, the recall for each sleep stage is comparatively low. Especially for sleep stages with small support, which are sleep stage 1, sleep stage 3 and sleep stage 4. The recall for sleep stage 1 is only 0.03, which is not acceptable.

With different variance of samples, the results have similar performance. The metrics for sleep stage 1, 3 and 4 are poor and the metrics for other sleep stages are good. When the variance is very low, the dataset itself is balanced. It means DBN itself is difficult to classify sleep stage 1, 3 and 4.



a) Low variance



b) High variance

Fig. 6.1 Confusion matrix of DBN

Fig 6.1 a) shows the confusion matrix of DBN with low variance. The performance is steady for all the sleep stages, but there’s no one with impressing high performance. Fig 6.1 b) shows the confusion matrix of DBN with high variance. DBN tends to predict sleep stages with small support to be sleep stage W, which may be caused by the imbalance of data. However, DBN has a pretty good performance to distinguish asleep state and awake state.

Comparing the performance in detail, when the variance is low, the performance is steady for all the sleep stages. However, there's no impressing highlight. When the variance is high, the performance is not steady for all the sleep stages. However, its performance for sleep stage W is extremely good.

6.1.2 Logistic Regression

The metrics of LR classification are showed in table 6.2-1 and table 6.2-2.

Table 6.2-1 Classification result of Logistic Regression (low variance)

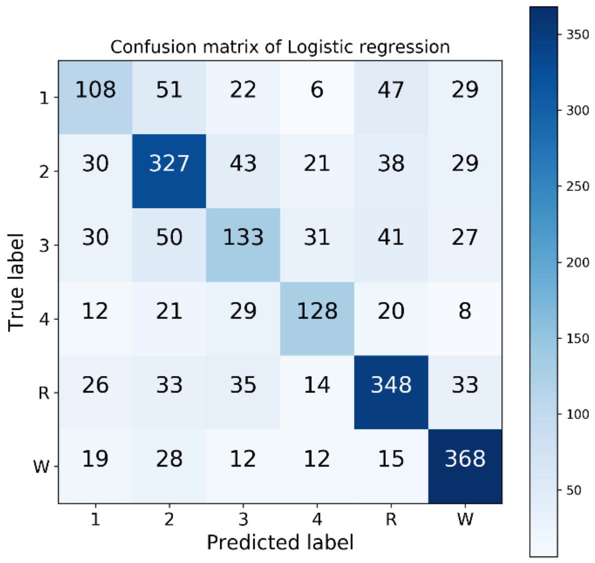
	Precision	Recall	f1-score	support
Sleep Stage 1	0.48	0.41	0.44	263
Sleep Stage 2	0.64	0.67	0.66	488
Sleep Stage 3	0.49	0.43	0.45	312
Sleep Stage 4	0.6	0.59	0.6	218
Sleep Stage R	0.68	0.71	0.7	489
Sleep Stage W	0.74	0.81	0.78	454
average/total	0.63	0.63	0.63	2224

Table 6.2-1 shows that with low variance, LR has a steady performance for each sleep stage. However, the recall for sleep stage 1 and sleep stage 3 is comparatively low.

Table 6.2-2 Classification result of Logistic Regression (high variance)

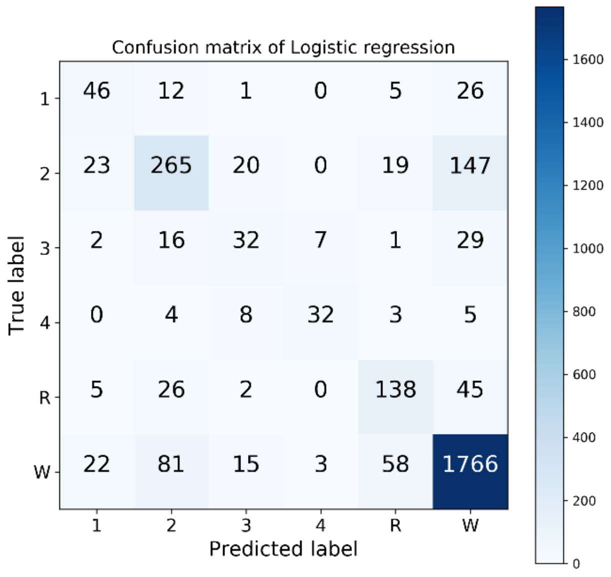
	Precision	Recall	f1-score	support
Sleep Stage 1	0.47	0.52	0.50	90
Sleep Stage 2	0.65	0.55	0.60	474
Sleep Stage 3	0.37	0.37	0.37	87
Sleep Stage 4	0.76	0.62	0.68	52
Sleep Stage R	0.59	0.63	0.61	216
Sleep Stage W	0.88	0.90	0.89	1945
average/total	0.79	0.79	0.79	2864

Table 6.2-2 shows that with high variance, LR still has a steady performance for each sleep stage. Except for the recall of sleep stage 3, all the metrics are acceptable. Comparing the above results, it could be found that the imbalance of samples does not have an obvious negative effect on the performance.



True \ Predicted	1	2	3	4	R	W
1	108	51	22	6	47	29
2	30	327	43	21	38	29
3	30	50	133	31	41	27
4	12	21	29	128	20	8
R	26	33	35	14	348	33
W	19	28	12	12	15	368

a) Low variance



b) High variance

Fig. 6.2 Confusion matrix of LR

Fig 6.2 a) shows the confusion matrix of LR with low variance. The performance is steady for all the sleep stages, but there’s no one with impressing high performance.

Fig 6.2 b) shows the confusion matrix of LR with high variance. The performance is steady for each sleep stage. The performance has not been influenced by the imbalance of data.

Comparing the results above, LR works better for the data with high variance.

6.1.3 Support Vector Machine

The metrics of SVM classification are showed in table 6.3-1 and table 6.3-2.

Table 6.3-1 Classification result of Support Vector Machine (low variance)

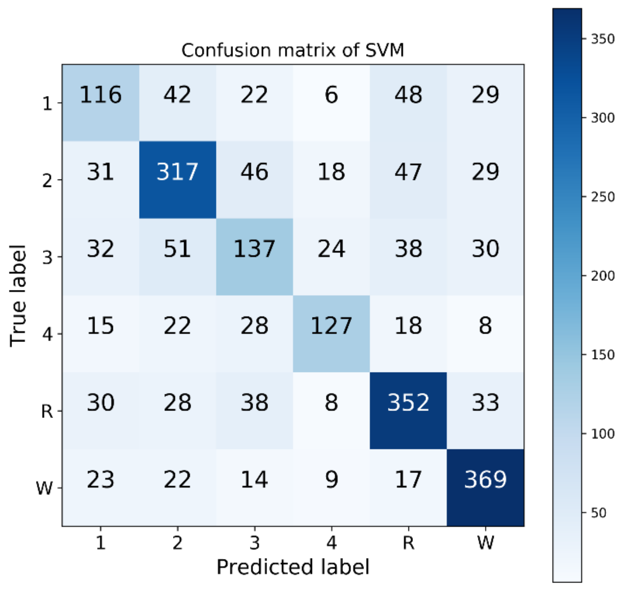
	Precision	Recall	f1-score	support
Sleep Stage 1	0.47	0.44	0.45	263
Sleep Stage 2	0.66	0.65	0.65	488
Sleep Stage 3	0.48	0.44	0.46	312
Sleep Stage 4	0.66	0.58	0.62	218
Sleep Stage R	0.68	0.72	0.7	489
Sleep Stage W	0.74	0.81	0.78	454
average/total	0.63	0.64	0.63	2224

Table 6.3-1 shows that with low variance, SVM has a steady performance for each sleep stage. However, the recall for sleep stage 1 and sleep stage 3 is comparatively low.

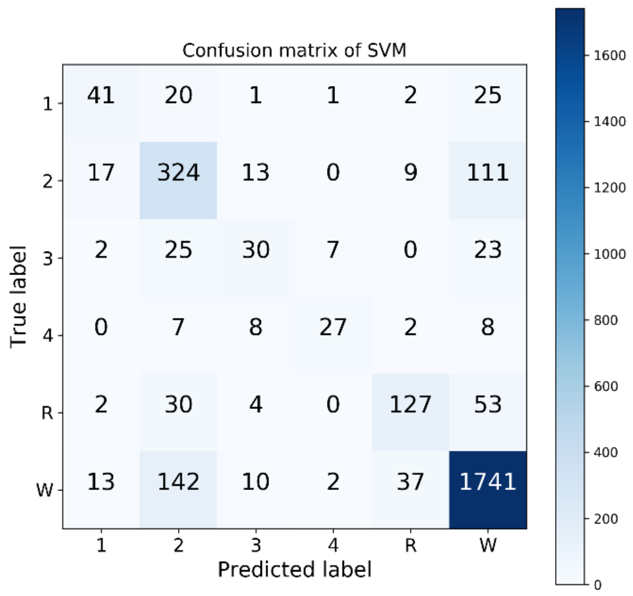
Table 6.3-2 Classification result of Support Vector Machine (high variance)

	Precision	Recall	f1-score	support
Sleep Stage 1	0.55	0.46	0.50	90
Sleep Stage 2	0.59	0.68	0.63	474
Sleep Stage 3	0.45	0.34	0.39	87
Sleep Stage 4	0.73	0.52	0.61	52
Sleep Stage R	0.72	0.59	0.65	216
Sleep Stage W	0.89	0.90	0.89	1945
average/total	0.80	0.80	0.80	2864

Table 6.3-2 shows that with high variance, SVM still has a steady performance for each sleep stage. Except for the recall for sleep stage 1 and 3, all the metrics are acceptable. Comparing the above results, it could be found that the imbalance of samples does not have an obvious negative effect on the performance.



a) Low variance



b) High variance

Fig. 6.3 Confusion matrix of SVM

Fig 6.3 a) shows the confusion matrix of SVM with low variance. The performance is steady for all the sleep stages, but there’s no one with impressing high performance.

Fig 6.3 b) shows the confusion matrix of SVM with high variance. The performance is steady for each sleep stage. The performance has not been influenced by the imbalance of data.

Comparing the results above, SVM works better for the data with high variance.

6.1.4 Gaussian Naïve Bayes

The metrics of GNB classification are showed in table 6.4-1 and table 6.4-2.

Table 6.4-1 Classification result of Gaussian Naïve Bayes (low variance)

	Precision	Recall	f1-score	support
Sleep Stage 1	0.15	0.14	0.15	263
Sleep Stage 2	0.31	0.06	0.09	488
Sleep Stage 3	0.26	0.07	0.11	312
Sleep Stage 4	0.14	0.53	0.22	218
Sleep Stage R	0.32	0.11	0.16	489
Sleep Stage W	0.29	0.51	0.37	454
average/total	0.26	0.22	0.18	2224

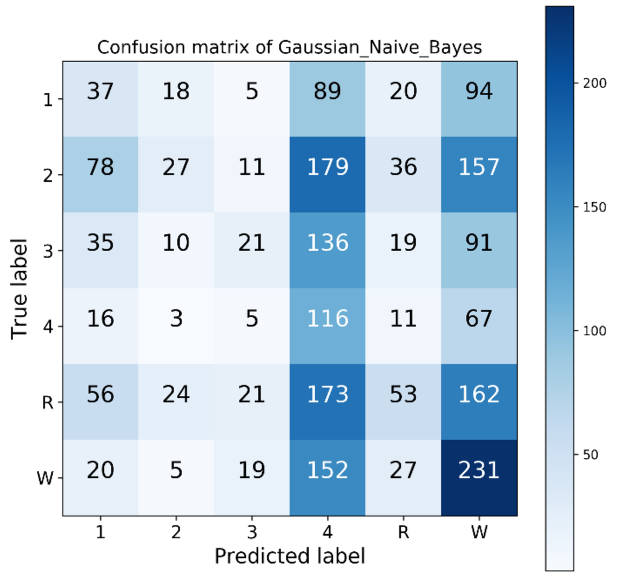
Table 6.4-1 shows that with low variance, GNB has a poor performance for all the sleep stages.

Table 6.4-2 Classification result of Gaussian Naïve Bayes (high variance)

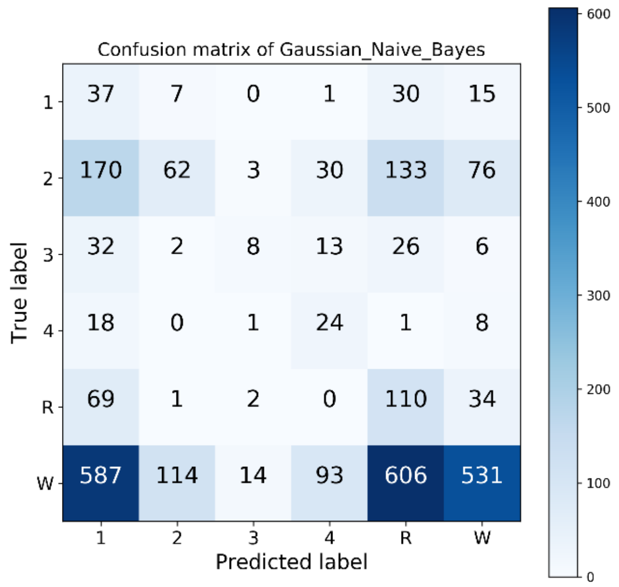
	Precision	Recall	f1-score	support
Sleep Stage 1	0.04	0.41	0.07	90
Sleep Stage 2	0.33	0.13	0.19	474
Sleep Stage 3	0.29	0.09	0.14	87
Sleep Stage 4	0.15	0.46	0.23	52
Sleep Stage R	0.12	0.51	0.2	216
Sleep Stage W	0.79	0.27	0.41	1945
average/total	0.62	0.27	0.33	2864

Table 6.4-2 shows that with high variance, GNB has a poor performance for all the

sleep stages. Except for the precision of sleep stage W, all the metrics are quite low. Comparing the results above, GNB has a poor performance for both two different datasets.



a) Low variance



b) High variance

Fig. 6.4 Confusion matrix of GNB

Fig 6.4 a) shows the confusion matrix of GNB with low variance. The performance is poor for all the sleep stages.

Fig 6.4 b) shows the confusion matrix of GNB with high variance. The performance is poor for each sleep stages.

Comparing the results above, GNB works poor and it should not be considered in the further study.

6.1.5 RBM-Logistic

The metrics of RBM-Logistic classification are showed in table 6.5-1 and table 6.5-2.

Table 6.5-1 Classification result of RBM-Logistic (low variance)

	Precision	Recall	f1-score	support
Sleep Stage 1	0	0	0	263
Sleep Stage 2	0	0	0	488
Sleep Stage 3	0	0	0	312
Sleep Stage 4	0	0	0	218
Sleep Stage R	0	0	0	489
Sleep Stage W	0.2	1	0.34	454
average/total	0.04	0.2	0.07	2224

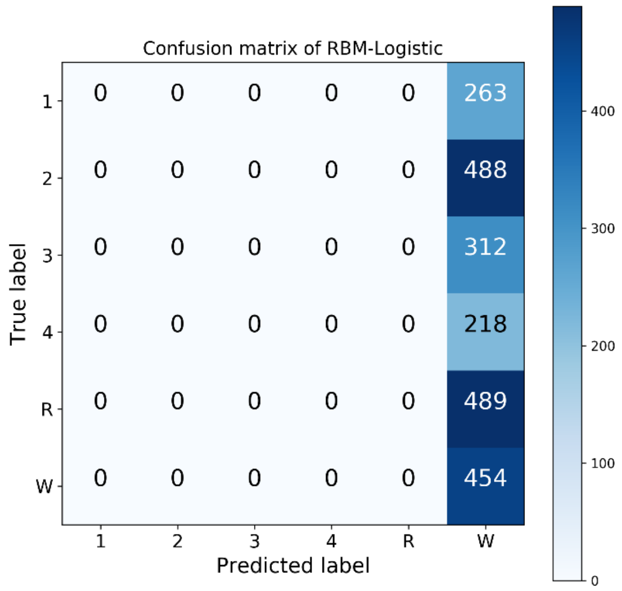
Table 6.5-1 shows that with low variance, RBM-Logistic has a poor performance for all the sleep stages.

Table 6.5-2 Classification result of RBM-Logistic (high variance)

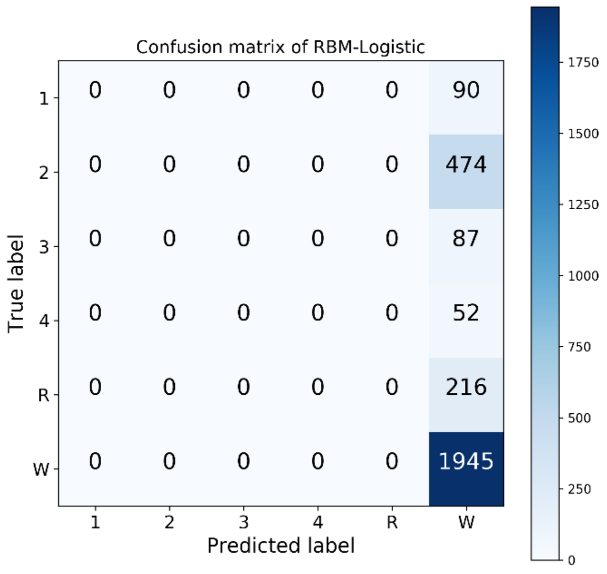
	Precision	Recall	f1-score	support
Sleep Stage 1	0	0	0	90
Sleep Stage 2	0	0	0	474
Sleep Stage 3	0	0	0	87
Sleep Stage 4	0	0	0	52
Sleep Stage R	0	0	0	216
Sleep Stage W	0.68	1	0.81	1945
average/total	0.46	0.68	0.55	2864

Table 6.5-2 shows that with high variance, RBM-Logistic has a poor performance for all the sleep stages. All the metrics are quite low.

Comparing the results above, RBM-Logistic has a poor performance for both two different datasets.



a) Low variance



b) High variance

Fig. 6.5 Confusion matrix of RBM-Logistic

Fig 6.5 a) shows the confusion matrix of RBM-Logistic with low variance. The performance is poor for all the sleep stages.

Fig 6.5 b) shows the confusion matrix of RBM-Logistic with high variance. The performance is poor for each sleep stages.

Comparing the results above, RBM-Logistic works poor and it should not be considered in the further study.

6.2 Comparison

After comparing all the classifiers above, it could be found that only DBN, LR and SVM have comparatively good performance. The performances of GNB and RBM-Logistic are so poor that they should not be considered in the further study. Besides, compared with using dataset with high variance, using dataset with low variance tends to provide a steady performance. However, its performance is not as impressing as that with high variance dataset when using DBN. Also, the change of variance has no obvious influence when using LR and SVM. So in further study, only the dataset with high variance will be used.

The average performance of these classifiers are narrowed down in table 6.6.

Table 6.6 Comparison between classifiers

	Precision	Recall	f1-score
DBN	0.80	0.81	0.77
Logistic Regression	0.79	0.79	0.79
SVM	0.80	0.80	0.80
GNB	0.62	0.27	0.33
RBM-Logistic	0.46	0.68	0.55

From table 6.6, it should be seen from the average value of metrics that DBN, Logistic Regression and SVM have an obvious advantage over GNB and RBM-Logistic. The performances of GNB and RBM-Logistic are very poor. Especially, the performance of RBM is unacceptable. The reason is that the pre-training for the network may not be suitable for the subsequent traditional supervised classification.

The average values of metrics of DBN, Logistic Regression and SVM are all very high and they are all similar. Details need to be considered separately. Fig 6.6 compares the difference more clearly.

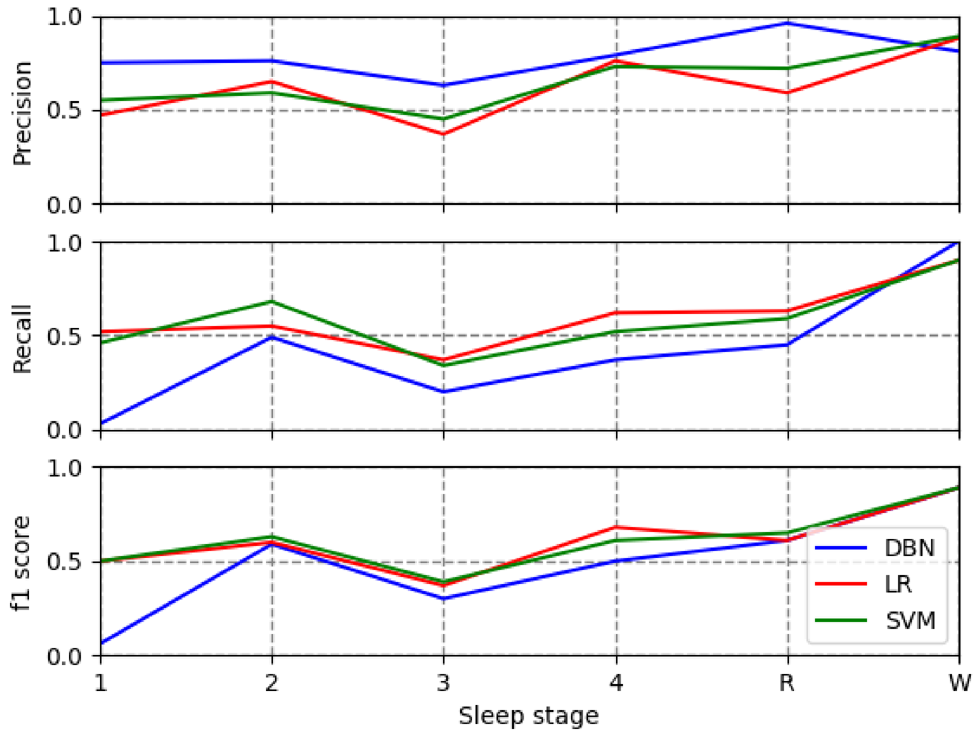


Fig. 6.6 Comparison of metrics for each sleep stage

Fig 6.6 shows that the overall performance of precision for DBN is better than that of LR and SVM. The precision for sleep stage W is slightly lower than LR and SVM, but they are similar. In contrast, the overall performances of recall for Logistic Regression and SVM are better than that of DBN. However, the low performance of DBN's recall focuses on the sleep stages except "Sleep stage W". Its recall for "Sleep stage W" is 100%, which is better than that of LR and SVM. It means DBN may confuse with different stages when a person is asleep, but it can classify the states of "asleep" and "awake" very clearly.

Besides, the performances of LR and SVM are similar. Considering the time-consuming property of SVM for large dataset, LR will be preferred for further study. For the f1 scores, traditional methods are slightly better than DBN, especially for sleep stage 1. The performance of DBN for sleep stage 2, R and W is better or similar as traditional methods.

In a word, traditional methods like LR and SVM is slightly better than deep learning methods like DBN. However, DBN also has a better performance when classifying sleep stage W, which is quite meaningful in reality. In addition, the performances of LR and SVM are similar and LR will be chosen for further study.

7. Improvement for DBN

As mentioned in part 6, DBN has the best performance for sleep stage W. Its performances for sleep stage 2 and R are similar as traditional methods like LR and SVM. However, its performance for other sleep stages, especially for sleep stage 1, is very poor. Traditional methods have a better performance for these sleep stages.

To improve the performance of DBN and keep its advantage for sleep stage W, we could combine DBN with traditional methods like LR. First classify sleep stage 1, sleep stage 3, sleep stage 4 and sleep stage others (contains sleep stage 2, R and W) using LR. Then use sleep stage others as test dataset to classify sleep stage 2, sleep stage R and sleep stage W with DBN.

In this way, the weakness of DBN for sleep stage 1, 3 and 4 will be made up by LR and DBN can still keep its advantage of sleep stage W.

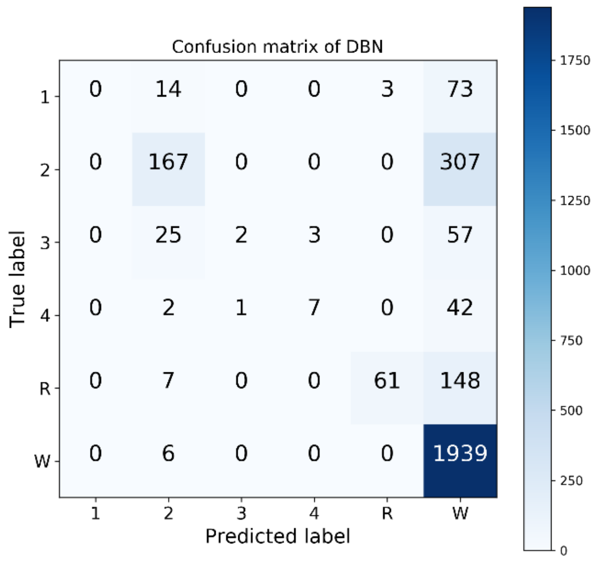
Table 7.1 Before improving

	Precision	**Recall**	**f1-score**	**support**
Sleep Stage 1	0.75	0.03	0.06	90
Sleep Stage 2	0.76	0.49	0.59	474
Sleep Stage 3	0.63	0.2	0.30	87
Sleep Stage 4	0.79	0.37	0.50	52
Sleep Stage R	0.96	0.45	0.61	216
Sleep Stage W	0.81	1.00	0.89	1945
average/total	0.80	0.81	0.77	2864

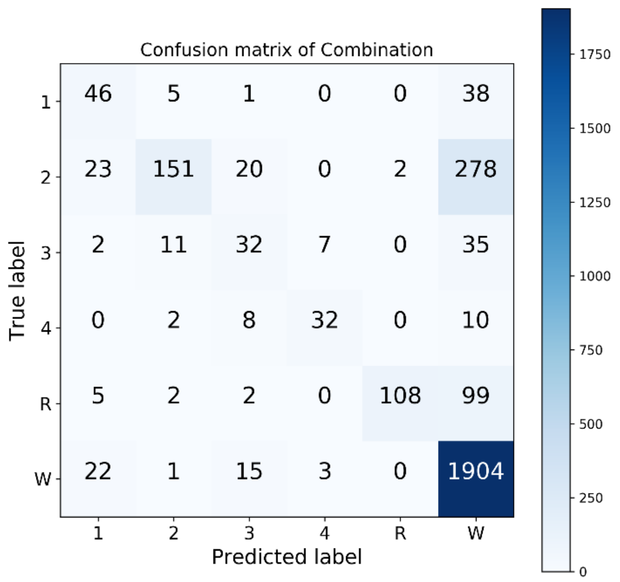
Table 7.2 After improving

	Precision	Recall	f1-score	support
Sleep Stage 1	0.47	0.52	0.49	90
Sleep Stage 2	0.83	0.43	0.57	474
Sleep Stage 3	0.45	0.44	0.44	87
Sleep Stage 4	0.77	0.63	0.69	52
Sleep Stage R	0.98	0.44	0.61	216
Sleep Stage W	0.83	0.98	0.9	1945
average/total	0.82	0.81	0.79	2864

Comparing table 7.1 and table 7.2, it could be found that, the performance of our classifier is steadier for each sleep stage. Except for sleep stage W, the recall for sleep stages has increased. Meanwhile, the performance of sleep stage W is still quite good after a little sacrifice.



a) Before improving



	1	2	3	4	R	W
1	46	5	1	0	0	38
2	23	151	20	0	2	278
3	2	11	32	7	0	35
4	0	2	8	32	0	10
R	5	2	2	0	108	99
W	22	1	15	3	0	1904

b) After improving

Fig. 7.1 Comparison for improvement

Fig 7.1 shows that after improving, the performance for sleep stage 1, 3 and 4 improves a lot. Meanwhile, the performances of sleep stage 2 and R have increased slightly. The performance of sleep stage W is still quite good.

To see the improvement more straightforwardly, the performance of each sleep stage before and after improving will be compared in fig 7.2.

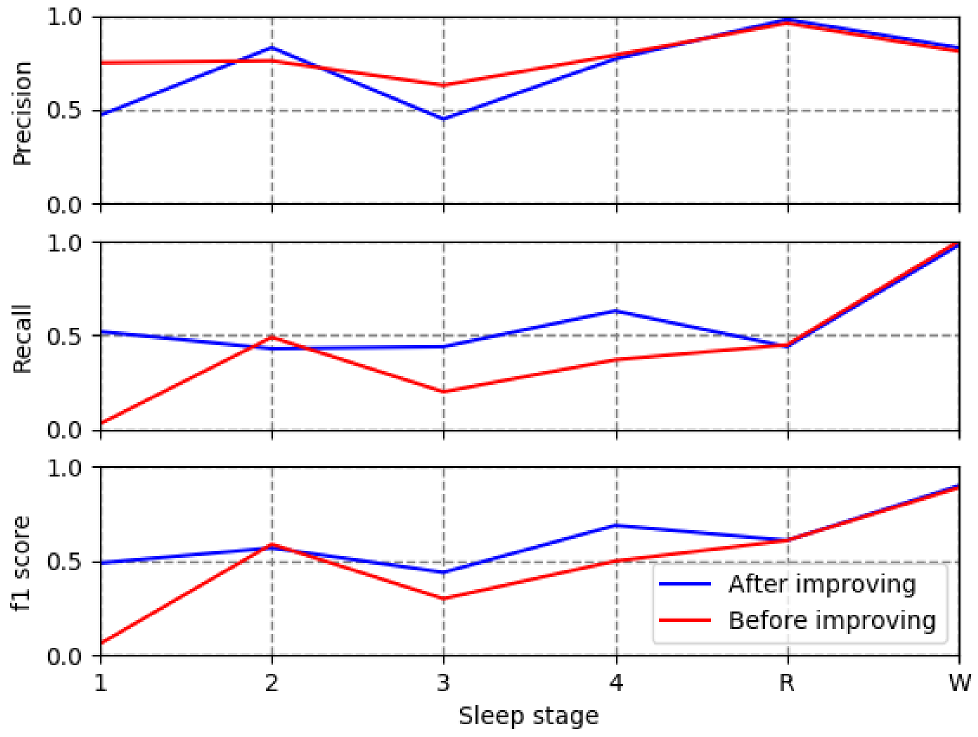


Fig 7.2 Comparison of metrics for improvement

Fig 7.2 shows that after improving, the performance of each sleep stage becomes steadier. Meanwhile, the impressing performance of sleep stage W has nearly no change.

The DBN after improving will be referred as improved DBN. Its overall performance is slightly worse than traditional methods like LR and SVM. Fig 7.3 compares the improved DBN with LR and SVM.

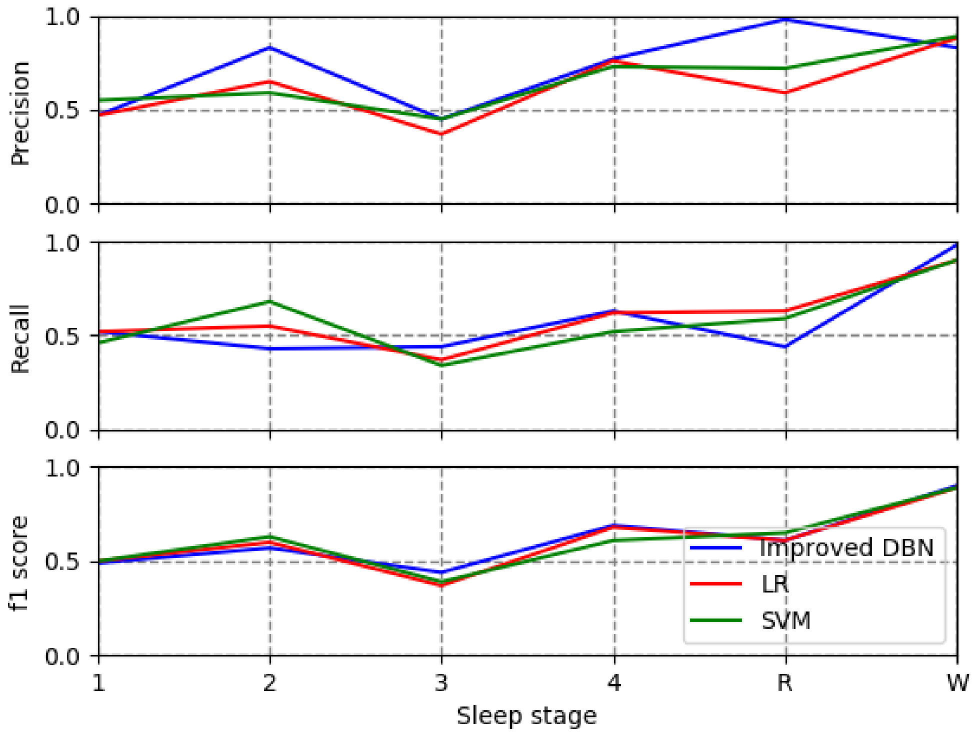


Fig 7.3 Comparison for different classifiers

Fig 7.3 shows that comparing LR and SVM, the performance of DBN has improved a lot. For precision, it even has an obvious advantage over traditional methods. The above result shows that the improvement this study used can efficiently enhance the performance of DBN.

8. Hidden Markov Model (HMM)

HMM is used to find the probabilities with maximum likelihood between sequences of samples. For instance, { sleep stage W, sleep stage W, sleep stage W, sleep stage 1, sleep stage 2, sleep stage 3, sleep stage 4, sleep stage R, sleep stage W, sleep stage W} is possible. However, a sequence like {sleep stage W, sleep stage 1, sleep stage W, sleep stage 2, sleep stage W, sleep stage 3, sleep stage W, sleep stage 4, sleep stage W, sleep stage R } is not likely to happen. It's too strange that sleep stage W happens between other sleep stages.

Using our prediction above as observation and using its true label as state, a HMM could be performed for our classifier.

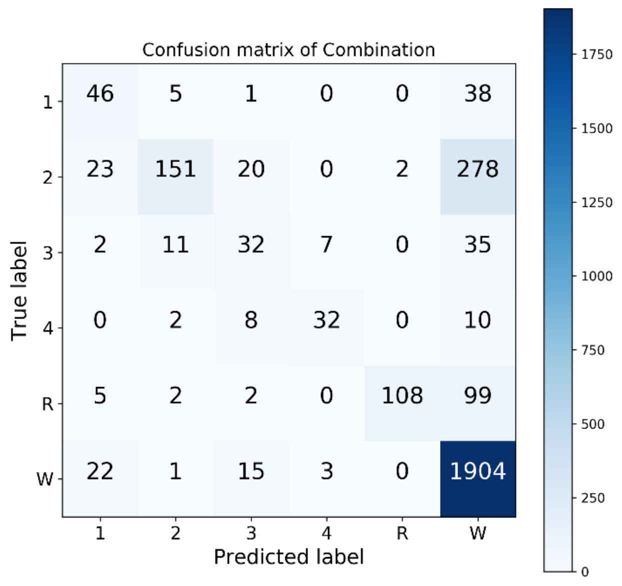
Table 8.1 Before HMM

	Precision	Recall	f1-score	support
Sleep Stage 1	0.47	0.52	0.49	90
Sleep Stage 2	0.83	0.43	0.57	474
Sleep Stage 3	0.45	0.44	0.44	87
Sleep Stage 4	0.77	0.63	0.69	52
Sleep Stage R	0.98	0.44	0.61	216
Sleep Stage W	0.83	0.98	0.9	1945
average/total	0.82	0.81	0.79	2864

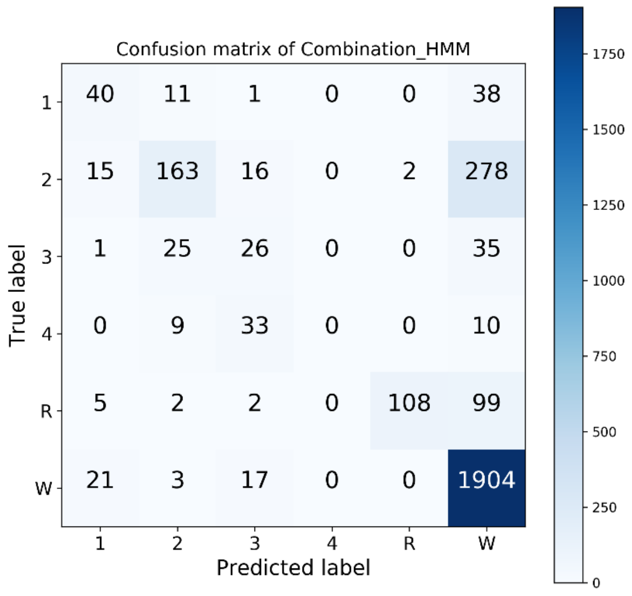
Table 8.2 After HMM

	Precision	Recall	f1-score	support
Sleep Stage 1	0.49	0.44	0.47	90
Sleep Stage 2	0.77	0.34	0.47	474
Sleep Stage 3	0.27	0.3	0.29	87
Sleep Stage 4	0	0	0	52
Sleep Stage R	0.98	0.5	0.66	216
Sleep Stage W	0.81	0.98	0.88	1945
average/total	0.77	0.78	0.75	2864

Comparing table 8.1 and table 8.2, after using HMM, the metrics for sleep stage 3 and 4 decrease a lot. Meanwhile, the overall performance becomes poorer.



c) Before HMM



d) After HMM

Fig. 8.1 Comparison for improvement

Fig 8.1 shows that after using HMM, there's no obvious improvement. The performances for sleep stage 3 and 4 even become poorer.

To see the decrease clearly, fig 8.2 will compare the metrics before and after using HMM.

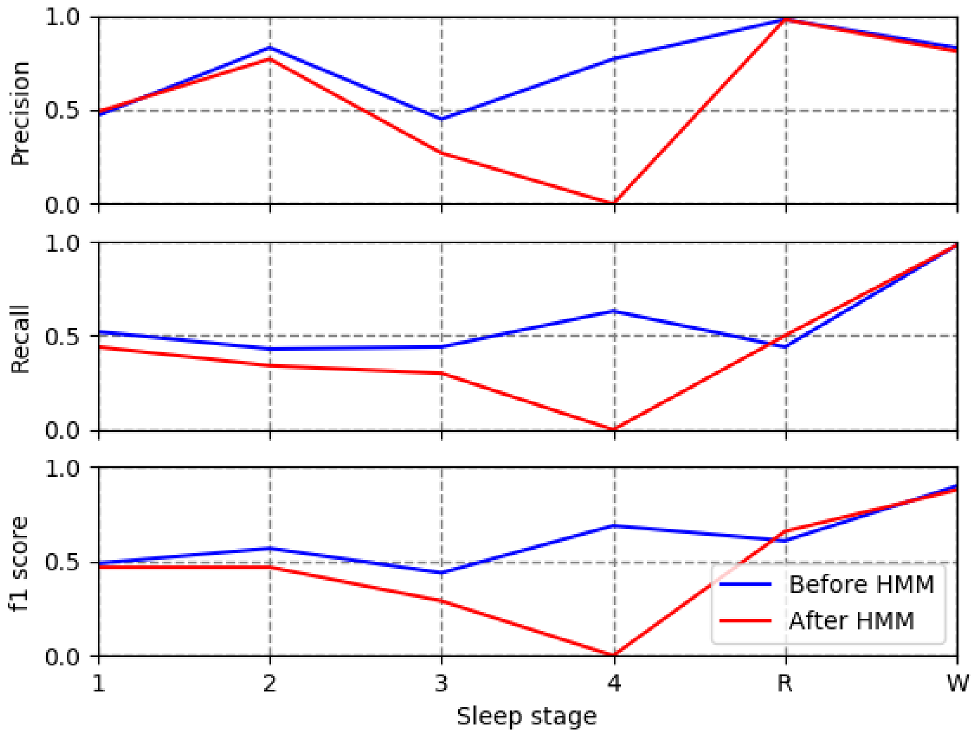


Fig 8.2 Comparison of metrics for HMM (high variance)

Fig 8.2 shows that after using HMM, the metrics for sleep stage 4 decrease a lot and they are not acceptable.

The reason that HMM can't improve the performance of our classifier and even lower its performance is that there's imbalance in our dataset. HMM calculates the maximum likelihood to form the transition probabilities matrix. If one of the classes is comparatively very small, then the transition probabilities for it will be comparatively very low and the result may go wrong.

Take the dataset referred as "low variance" in part 6 as example, fig 8.3 shows the metrics before and after using HMM.

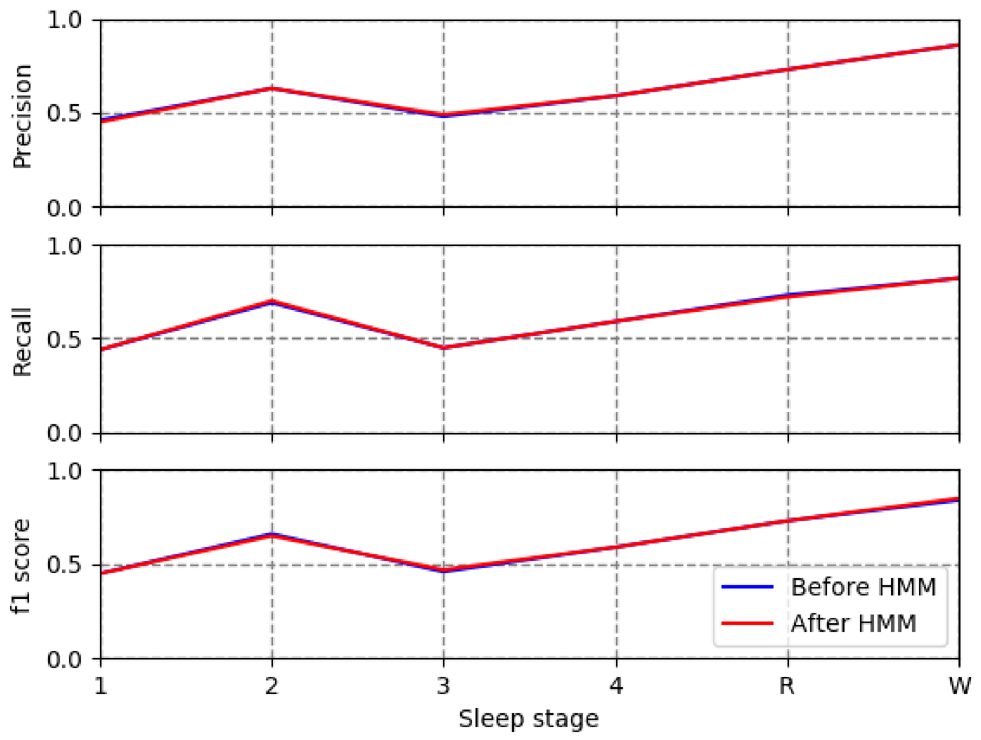


Fig 8.3 Comparison of metrics for HMM (low variance)

When the dataset is balanced, there will be no obvious decrease for the performance. However, there's also no obvious improvement.

To conclude, HMM has strict requirement for balanced dataset, which could not always be met for real EEG signals. Even the data is balanced, there's no obvious improvement. So HMM will not be recommended to classify sleep stages in this study.

9. Resources and experiment environment

9.1 Hardware Specification:

- win 10: Processor: intel core i5-5200U; Memory: 8GB

9.2 Software Specification:

- Data processing part: The origin format of dataset is ".edf". "Polyman" [17] is used to visualize the signals of this format. “pyedflib-master” package [18] is used to process the original dataset of this format.

- Machine learning part: The DBN part employs “deep-belief-network-master” package [16]. The HMM part employs “hmmlearn-master” package [19]. Other machine learning part employs “scikit_learn-0.19.0”.

10. Conclusion

Classifying sleep stages using machine learning methods is very meaningful to study sleep activity. EEG signals can be used for classification. According to R&K criterion, sleep stages can be separated into stage 1, stage 2, stage 3, stage 4, stage R and stage W. After comparison, the train and test data is split from all the epochs and the ratio between different sleep stages is just as the original data source. Then four classifiers have been compared with Deep Belief Networks. The result shows that traditional methods have a slightly better performance. To improve the performance of Deep Belief Networks, Logistic Regression is combined with it. The improvement successfully enhances the performance. In addition, Hidden Markov Model is tested to modify the prediction and the conclusion shows that Hidden Markov Model is not recommended for this project. The conclusions of this study can be narrowed down into five points.

- After comparing two ways to split train and test data, the way that firstly collect all the samples into one dataset and then split train and test data from it will be adopted.
- Using variance to measure the extent of imbalance of dataset, dataset with high variance will be used for classification.
- Traditional methods like LR and SVM have better overall performance to classify the sleep stages, but DBN works better to distinguish awake state from asleep state.
- By combining DBN with LR, the performance of DBN can be improved.
- HMM is not recommended in this study.

References

[1] Sun, S. and Zhang, C., 2006. Adaptive feature extraction for EEG signal classification. *Medical and Biological Engineering and Computing*, *44*(10), pp.931-935.

[2] Güneş, S., Polat, K. and Yosunkaya, Ş., 2010. Efficient sleep stage recognition system based on EEG signal using k-means clustering based feature weighting. *Expert Systems with Applications*, *37*(12), pp.7922-7928.

[3] Li, X., Cui, L., Tao, S., Chen, J., Zhang, X. and Zhang, G.Q., 2017. HyCLASSS: A Hybrid Classifier for Automatic Sleep Stage Scoring. *IEEE Journal of Biomedical and Health Informatics*.

[4] Supratak, A., Dong, H., Wu, C. and Guo, Y., 2017. DeepSleepNet: a Model for Automatic Sleep Stage Scoring based on Raw Single-Channel EEG. *arXiv preprint arXiv:1703.04046*.

[5] Li, K., Li, X., Zhang, Y. and Zhang, A., 2013, December. Affective state recognition from EEG with deep belief networks. In *Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on* (pp. 305-310). IEEE.

[6] Xu, H. and Plataniotis, K.N., 2016, September. Affective states classification using EEG and semi-supervised deep learning approaches. In *Multimedia Signal Processing (MMSP), 2016 IEEE 18th International Workshop on* (pp. 1-6). IEEE.

[7] Xia, B., Li, Q., Jia, J., Wang, J., Chaudhary, U., Ramos-Murguialday, A. and Birbaumer, N., 2015, July. Electrooculogram based sleep stage classification using deep belief network. In *Neural Networks (IJCNN), 2015 International Joint Conference on* (pp. 1-5). IEEE.

[8] Physionet.org. (2017). *The Sleep-EDF Database [Expanded]*. [online] Available at: https://physionet.org/physiobank/database/sleep-edfx/ [Accessed 10 Nov. 2017].

[9] Kemp, B., Zwinderman, A.H., Tuk, B., Kamphuisen, H.A. and Oberye, J.J., 2000. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG. *IEEE Transactions on Biomedical Engineering*, *47*(9), pp.1185-1194.

[10] Rechtschaffen, A., 1968. A manual of standardized terminology, techniques and

scoring system for sleep stages of human subjects. *Public health service*.

[11] En.wikipedia.org. (2017). *Fourier transform*. [online] Available at: https://en.wikipedia.org/wiki/Fourier_transform [Accessed 10 Nov. 2017].

[12] Schoonjans, F. (2017). *Logistic regression*. [online] MedCalc. Available at: https://www.medcalc.org/manual/logistic_regression.php [Accessed 10 Nov. 2017].

[13] En.wikipedia.org. (2017). *Support vector machine*. [online] Available at: https://en.wikipedia.org/wiki/Support_vector_machine [Accessed 10 Nov. 2017].

[14] Brownlee, J. (2017). *Naive Bayes for Machine Learning - Machine Learning Mastery*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/naive-bayes-for-machine-learning/ [Accessed 10 Nov. 2017].

[15] En.wikipedia.org. (2017). *Hidden Markov model*. [online] Available at: https://en.wikipedia.org/wiki/Hidden_Markov_model [Accessed 10 Nov. 2017].

[16] GitHub. (2017). *albertbup/deep-belief-network*. [online] Available at: https://github.com/albertbup/deep-belief-network [Accessed 10 Nov. 2017].

[17] Physionet.org. (2017). *Polyman for MS-Windows*. [online] Available at: https://physionet.org/physiobank/database/sleep-edfx/Polyman/ [Accessed 10 Nov. 2017].

[18] GitHub. (2017). *holgern/pyedflib*. [online] Available at: https://github.com/holgern/pyedflib [Accessed 10 Nov. 2017].

[19] GitHub. (2017). *hmmlearn/hmmlearn*. [online] Available at: https://github.com/hmmlearn/hmmlearn [Accessed 10 Nov. 2017].