Jurnal Ilmiah KURSOR Menuju Solusi Teknologi Informasi

Vol. 8, No. 4, Desember 2016

ISSN 0216 – 0544
e-ISSN 2301– 6914

COMBINATION DEEP BELIEF NETWORKS AND SHALLOW CLASSIFIER FOR SLEEP STAGE CLASSIFICATION

[a]Intan Nurma Yulita, [b]Rudi Rosadi, [c]Sri Purwani, [d]Rolly Maulana Awangga
[a,b,c]Universitas Padjadjaran, Sumedang Indonesia 45363
[d]Politeknik Pos Indonesia, Bandung Indonesia 40151
E-mail: intan.nurma@unpad.ac.id

Abstract

In this research, it is proposed to use Deep Belief Networks (DBN) in shallow classifier for the automatic sleep stage classification. The automatic classification is required to minimize the evaluation of Polysomnography because it needs more than two days for analysis manually. Thus the automatically mechanism is required. The Shallow classifier used in this research includes Naïve Bayes (NB), Bayesian Networks (BN), Decision Tree (DT), Support Vector Machines (SVM), and K-Nearest Neighbor (KNN). The analysis compared each methods in shallow classifier before and after the classifier were combined with DBN. The results shown that many combination by using the shallow classifiers and DBN had increased. The experiments that have been done indicated a significant increase of Naive Bayes after being combined with DBN. The high-level features generated by DBN are proven to be useful in helping Naive Bayes' performance. On the other hand, the combination of KNN with DBN shows a decrease because high-level features of DBN make it harder to find neighbors that optimize the performance of KNN.

Keywords: Deep Belief Networks, Sleep Apnea, Sleep Stage Classification, Shallow Classifier

INTRODUCTION

Sleep apnea is a serious sleep disorder where there are some stops when someone breaths in sleep. As a result, the organs, especially the brain, may not get enough oxygen, as well as poor sleep quality, which can make patients feel tired the next day. Unconsciously, sleep apnea can be a silent killer.

Therefore, it is necessary to do an examination in a sleep laboratory using Polysomnography To diagnose a sleep disorder. The test required as a first step in determining sleep disturbance therapy. This test works by recording sleeping conditions and stopping the patient's breathing. From the review can be known the quality of sleep, the type and degree of sleep disorders seen from the number of stopping breath per minute and duration of breathing stops, and decreased oxygen levels that occur when the patient fell asleep.

This test requires many cables to be attached to some parts of the body. Completeness of the device and the sensor must also meet the requirements of the standards contained in the rules of the American Academy of Sleep Medicine. Full sleep recording includes recording: Snoring, Air Breath, Chest and abdominal movement, Oxygen level (SpO2), Sleeping position Heart rhythm (ECG), Sleep brainwaves (EEG), Eye movement (EOG), Muscle activity EMG) on the chin, and legs [1].

Furthermore, the recording results will be read and assessed by a qualified physician in the field of sleep, and then made the report. This process usually lasts between 2 to 3 days. Then the patient can meet with his doctor again to take the results.

With the length of time to analyze the recording, then data mining will be useful to reduce the time available [2]. Therefore, in this study, sleep stage classification is done with the approach of data mining. This classification is expected to accelerate decision making on the patient's condition.

Several studies have tested some methods for sleep stage classification such as Neural Networks [3], Support Vector Machine [4], and K-Means Clustering [5]. However, this study offers the use of Deep Belief Networks as a feature representation for the Shallow Classifier, and compares among shallow classifier and their combinations. The shallow classifiers used are Naïve Bayes, Decision Tree, Bayesian Networks, K-Nearest Neighbor, and Support Vector Machine. The classifiers are some frequently used method for solving classification problems. So with Deep Belief Networks, then the methods can produce better performance.

METHODOLOGY

Classification consists of 2 process steps. First is learning (training phase), where the classification algorithm is made to analyze training data and then represented in the classification rule. The second process is the classification, where the test data is used to estimate the accuracy of the classification rule. In this study, several methods used for sleep stage classification, namely:

Deep Belief Networks (DBN)

The DBN implemented in this research is based on Restricted Boltzmann Machine (RBM). RBM is a variant of the Boltzmann machine method with restrictions, in which the visible layer and the hidden layer [6]. Visible layers and hidden layers are connected but between two visible or hidden layers are not attached to each other [7]. The visible unit of RMB is the input to the hidden part of the network, which represents the feature detector. It is what will do to this research by making DBN as feature representation.

Naïve Bayes (NB)

Naive Bayes uses probability theory as the basis of the theory [8]. Naive Bayes has a high level of speed and accuracy when applied to great databases. In determining the class of a data at the time of classification, all labels tested on the data by using Bayes theorem. The class that has the highest probability value becomes predicted from the c. Bayes theorem described in Equation (1) and Equation (2).

$$P\ (C_i |\ X) = \frac{P\ (X|C_i).P\ (C_i)}{P\ (X)} \tag{1}$$

$$P\ (X|C_i) = \prod_{t=1}^{n} P\ (X_t|C_i) \tag{2}$$

Where
X: sample data
C_i: target class (C_1, C_2, …, C_5)
X_t: attribute of sample data

Decision Tree (DT)

The decision tree is a classification tree consisting of nodes representing attributes, and leaves representing a particular class [9]. The top part of a decision tree is called root and is the most important attribute to the class determination. The decision tree built by looking for the most important features. The selection of an attribute as a node, either root (root) or internal node based on the highest Gain value of the existing attributes [10]. Gain and entropy calculations described in Equation (3) and Equation (4).

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i) \quad (3)$$

$$Entropy(S) = \sum_{i=1}^{n} -pi * log_2\, pi \quad (4)$$

Where:
S : The set of cases
A : Attribute
n : Number of partitions attributes A
$|S_i|$: Number of cases on i^{th} partition
|S| : Number of cases in S
pi : The proportion of Si against S

Attributes selected to the top node. To get a complete tree structure, then this process repeated so that all data has classified with the tree that has built. Testing of new data done with a top-down search strategy for the solution.

Bayesian Network (BN)

Bayesian Networks (BN) is a probabilistic data modeling method that represents a set of variables and conditional interdependencies through Directed Acyclic Graph. Although both derived from the Bayes theorem, there is a difference between the NB and the BN. The significant difference between the two methods lies in the presence or absence of interrelationships between variables. This connectivity is ignored NB while not on the BN. The classifier implements a joint probability distribution [11]. Join probability distribution is the likelihood of joint appearance for all possible/combination of values that exist between X and Y. So in the method is known chain rules so that a structure is likely to occur from N node/variable.

K-Nearest Neighbor (KNN)

K-nearest neighbor (KNN) is an instance-based learning group. This algorithm is also one of the lazy learning techniques. KNN did by searching k-group objects in the closest training data (similar) to objects in new data or data testing. The working principle of K-Nearest Neighbor (KNN) is to find the closest distance between the data to evaluate with the k nearest neighbor in the training data [12].

The calculation of the distance between the data with each other can be done by the formula Euclidean distance, as described in the Equation (5).

$$d_i = \sqrt{\sum_{i=1}^{m} (x_{2i} - x_{1i})^2} \quad (5)$$

Where
x_1 = sample data
x_2 = test data
i = data variables
d = distance
m = data dimension

Support Vector Machine (SVM)

The primary target of SVM is to find the best hyperplane to obtain maximum margin size [13]. The margin is the distance between the hyperplane and the nearest point of each class. The closest point is called the support vector. SVM usage is limited to a small problem because SVM training algorithms tend to be slow, complex, and difficult to implement. Therefore, Sequential Minimal Optimization (SMO) is developed to provide solutions to optimization problems. At each stage, SMO selects two Lagrange multipliers α i to be optimized together, finds the most optimal value for the Lagrange multiplier, and renews SVM with the new optimal value.

With some such methods, it is proposed by Deep Belief Networks (DBN) to find represented features as inputs on the shallow classifier. The sleep stage classification mechanism proposed in this study shown in Figure 1.

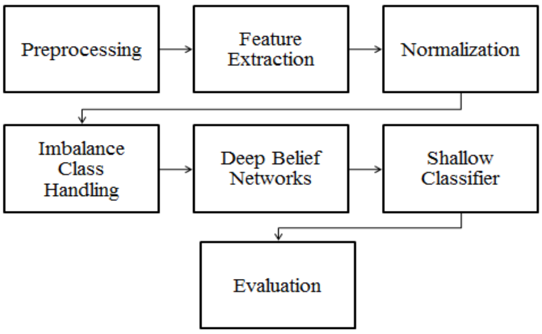

Figure 1. Flowchart

Table 1. Features List

Features	Features (contd)
EEG delta	EMG gamma
EEG theta	EMG median
EEG alpha	EOG correlation
EEG beta	EEG kurtosis
EEG gamma	EOG kurtosis
EOG delta	EMG kurtosis
EOG theta	EOG standard deviation
EOG alpha	EEG entropy
EOG beta	EOG entropy
EOG gamma	EMG entropy
EMG delta	EEG spectral mean
EMG theta	EOG spectral mean
EMG alpha	EMG spectral mean
EMG beta	EEG fractal exponent

The steps of sleep stage classification were:

a. Preprocessing
The first step was the processing of raw data. The input of this step was sleep data in Polysomnography. The processes included up-sampling, down-sampling, and filtering (notch and bandpass filtering).

b. Feature extraction
The output of the preprocessing stage were extracted to 28 features with the details in Table 1.

c. Normalization
The normalization process scaled back the range of values of some features with the aim that the previous range difference did not affect the feature representation or classification process.

d. Imbalance class handling
Five sleep stages used in this study, but there was a difference between some stages with other stages. The mechanism in this research was to equalize the number of samples from all classes to the smallest class (minor class). Thus, data trimming occurred from non-minor classes.

e. Deep Belief Networks
The DBN used in this study consisted of three layers where the last layer were supervised. The output of this stage was 28 features of the data and would to be converted into only five features. These five new features were called represented features.

f. Shallow classifier
The output of DBN with five-dimensional data of the attributes became the input for shallow classifiers. Then the five classifiers built the model based on the data provided except KNN. KNN is a lazy classifier, so it did not require a model.

g. Evaluation
The method were evaluated based on precision, recall, and F-measure to see the performance of the proposed method. The precision is the number of sleep stages that are relevant to the number of sleep stages obtained. It is related to the ability of a system to find the appropriate sleep stage. The recall is linked to the capacity of the system to acquire the proper sleep phase, whereas precision is related to the system's ability to not classify irrelevant sleep stages while F-Measure is a combination of precision and recall. The dataset examined in this study comes from https://www.physionet.org/pn3/ucddb/. The final evaluation were the comparison among the shallow classifier, and also their combination with DBN.

RESULT AND DISCUSSION

This study offered Deep Belief Networks as feature representation on the shallow classifier. There was also a comparison of performance without DBN To see the effect of feature representation; Table 2 shows the results. The highest performance of DBN was only reached by 0.04.

Table 2. Results

Methods	Precision	Recall	F-Measure
DT	0.54	0.55	0.54
DBN+DT	0.55	0.57	0.56
KNN	0.53	0.54	0.53
DBN+KNN	0.52	0.53	0.52
BN	0.54	0.55	0.54
DBN+BN	0.57	0.57	0.57
SVM	0.57	0.58	0.57
DBN+SVM	0.57	0.58	0.58
NB	0.53	0.51	0.51
DBN+NB	0.55	0.57	0.55

Precision

Based on Figure 2, DBN increased the value of precision when implemented as a feature representation on DT, BN, and NB. The biggest increase occurred in NB and the smallest in BN. It indicated that inter-attribute dependence occurs. If the connectivity of NB without DBN was not able to present, then it arised by using BN. However, when DBN had been implemented on NB, it had been handled during the DBN process as a feature representation. On the other hand, DBN did not give any effect in the creation of SVM hyperplanes, so the precision did not change. Also, the precision of KNN had decreased. It was because KNN works with the principle of finding the nearest neighbor, even though it was in a different class. If the DBN combined with KNN, then the closest neighbor of a forced data should be in the same class.

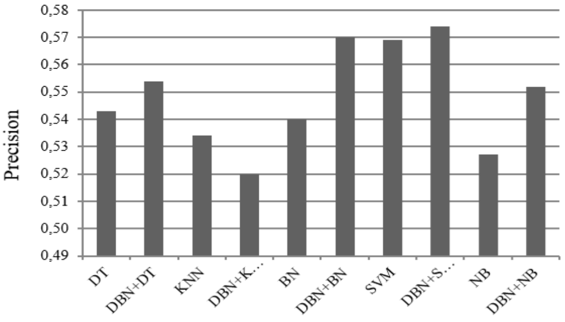

Figure 2. Precision

Recall

The recall in this study shown in Figure 3. Similar to precision, recall increases occur in DT, BN, and NB. While SVM did not increase, even KNN decreased recall. If compared to precision, the recall value was higher except for NB. However, NB experienced a more significant increase of 0.06 with DBN as feature representation.

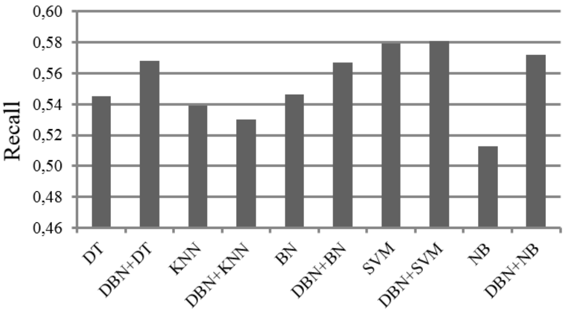

Figure 3. Recall

F-Measure

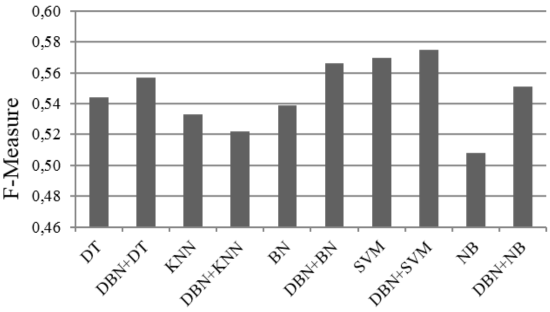

Figure 4. F-Measure

Figure 4 shows that most methods had been improved with the implementation of DBN as a feature representation. F-Measure DBN + DT was greater than 0.02 than DT, DBN + BN was 0.03, DBN + SVM was 0.01, and DBN + NB was 0.04. The decrease of F-Measure occurred at KNN that was equal to 0.01.

CONCLUSION

Based on the results obtained, it is known that:

1) There was an increase in the performance of both precision, recall, and F-Measure on some shallow classifier (NB, BN, DT).
2) KNN shown performance reduction. The represented features of DBN made it harder to find the right neighbors in KNN for classification.
3) Based on F-Measure, the performance of SVM had been increased, but a stable value in precision and recall occurred when this method combined with DBN.
4) NB had the sharpest increase due to the dependency properties between attributes that occurred in the data could be handled when the DBN implemented.
5) Performance improvements occur during the implementation of DBN and shallow classifier in small numbers.

Thus, it is suggested in subsequent research to implement DBN on sequence classifier data as performed by Hidden Markov Models and Long Short-Term Memory.

REFERENCES

[1] K. A. I. Aboalayon, M. Faezipour, W. S. Almuhammadi, and S. Moslehpour, “Sleep Stage Classification Using EEG Signal Analysis: A Comprehensive Survey and New Investigation,” *Entropy*, vol. 18, no. 9, p. 272, Aug. 2016.

[2] V. Bajaj and R. B. Pachori, “Automatic classification of sleep stages based on the time-frequency image of EEG signals,” *Comput. Methods Programs Biomed.*, vol. 112, no. 3, pp. 320–328, 2013.

[3] F. Ebrahimi, M. Mikaeili, E. Estrada, and H. Nazeran, “Automatic sleep stage classification based on EEG signals by using neural networks and wavelet packet coefficients,” in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2008, pp. 1151–1154.

[4] S. Khalighi, T. Sousa, D. Oliveira, G. Pires, and U. Nunes, “Efficient feature selection for sleep staging based on maximal overlap discrete wavelet transform and SVM,” in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, 2011, pp. 3306–3309.

[5] S. Güneş, K. Polat, and Ş. Yosunkaya, “Efficient sleep stage recognition system based on EEG signal using k-means clustering based feature weighting,” *Expert Syst. Appl.*, vol. 37, no. 12, pp. 7922–7928, 2010.

[6] J. Zhang, Y. Wu, J. Bai, and F. Chen, “Automatic sleep stage classification based on sparse deep belief net and combination of multiple classifiers,” *Trans. Inst. Meas. Control*, vol. 38, no. 4, pp. 435–451, 2016.

[7] V. Nair and G. E. Hinton, “3D object recognition with deep belief nets,” in *Advances in neural information processing systems*, 2009, pp. 1339–1347.

[8] H. J. Escalante, E. F. Morales, and L. E. Sucar, “A naive bayes baseline for early gesture recognition,” *Pattern Recognit. Lett.*, vol. 73, pp. 91–99, 2016.

[9] J. Chen, H. M. Le, P. Carr, Y. Yue, and J. J. Little, “Learning online smooth predictors for realtime camera planning using recurrent decision trees,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4688–4696.

[10] G. Lin, C. Shen, Q. Shi, A. Van den Hengel, and D. Suter, “Fast supervised hashing with decision trees for high-dimensional data,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1963–1970.

[11] S. Balbi, F. Villa, V. Mojtahed, K. T. Hegetschweiler, and C. Giupponi, “A spatial Bayesian network model to assess the benefits of early warning for urban flood risk to people,” *Nat. Hazards Earth Syst. Sci. Discuss.*, vol. 3, pp. 6615–6649, 2015.

[12] Y. Elmehdwi, B. K. Samanthula, and W. Jiang, “Secure k-nearest neighbor query over encrypted data in outsourced environments,” in *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, 2014, pp. 664–675.

[13] X. Huang, L. Shi, and J. A. Suykens, “Support vector machine classifier with pinball loss,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 984–997, 2014.