Combining Deep Belief Networks and Bidirectional Long Short-Term Memory

Case Study: Sleep Stage Classification

Intan Nurma Yulita[ab1], Mohamad Ivan Fanany[a2], Aniati Murni Arymurthy[a3]
[a]Faculty of Computer Science, Universitas Indonesia
Depok, Indonesa
[2]ivan@cs.ui.ac.id
[3]aniati@cs.ui.ac.id
[b]Department of Computer Science, Universitas Padjadjaran
Sumedang, Indonesia
[1]intan.nurma@unpad.ac.id

Abstract— **This paper proposes a new combination of Deep Belief Networks (DBN) and Bidirectional Long Short-Term Memory (Bi-LSTM) for Sleep Stage Classification. Tests were performed using sleep stages of 25 patients with sleep disorders. The recording comes from electroencephalography (EEG), electromyography (EMG), and electrooculography (EOG) represented in signal form. All three of these signals processed and extracted to produce 28 features. The next stage, DBN Bi-LSTM is applied. The analysis of this combination compared with the DBN, DBN HMM (Hidden Markov Models), and Bi-LSTM. The results obtained that DBN Bi-LSTM is the best based on precision, recall, and F1 score.**

Keywords—Deep Belief Networks; Long Short-Term Memory, Bidirectional Long Short-Term Memory; Hidden Markov Models; Sleep Stage Classification

I. Introduction

Polysomnogram is a sleep recording data used by a doctor to analyze a patient's with a sleep disorder. Polysomnogramusually derived from electroencephalography (EEG), electromyography (EMG), and electrooculography (EOG), which represented in the form of signals. Analysis of patient with sleep disorders can do by determining the sleep stage of each segment in the signal. So to facilitate the work of doctors, then it required a system that classifies automatically from these signals.

The approach that can use for sleep stage classification is through the utilization of the following information from data signals. The commonly used method is Hidden Markov Models (HMM). The approach was developed for speech recognition [1] but has grown widely for the application of action recognition [2], and speech recognition [3]. The other hand, Martin Langkvist et.al also developed it in Sleep Stage Classification [4]. In the study, they performed three mechanisms of sleep stage classification: Gaussian Observation HMM (GOHMM) with feature extraction, Deep Belief Networks-HMM (DBN HMM) with feature extraction, and DBN HMM with raw-data. The best mechanism obtained through a combination of DBN HMM with feature extraction.

But HMM with its success in many applications still has its drawbacks. One of them is the limited storage of information within the hidden state. Recurrent Neural Networks (RNN) developed To overcome the existing deficiencies. The most important thing in RNN is the existence of distributed hidden states that can store more information and update capabilities of hidden-states to the information they have. Unfortunately, training will be tough to do. One type of RNN that can use is Long Short-Term Memory (LSTM). LSTM is one of the particular types of RNN that can be trained using Gradient Descent [5]. LSTM formed from some memory blocks consisting of several gates. Gates has a task in the storage settings of memory cells. But it has a direction so that to maximize the functionality of LSTM, LSTM combined with bi-directional networks into Bidirectional LSTM (Bi-LSTM) [6].

With the advantages of Bi-LSTM then this research applies it to Sleep Stage Classification. Furthermore, to improve performance, Bi-LSTM is combined with DBN. The structure of the classification is formed from DBN on the down-level and Bi-LSTM on the top level so that the output of DBN becomes input from Bi-LSTM. DBN implementation aims to produce artificial features.

DBN as one of the generative models of Deep Learning consists of several layers of Restricted Boltzmann Machine (RBM). The top layer of DBN uses a classifier, for example, softmax classifier. RBM produces new features in its processing. It distinguishes it from the generative model of Deep Learning, such as Convolutional Neural Networks. The incorporation of DBN on Bi-LSTM is done by using artificial features obtained at the RBM stage. However, in this study, the artificial features taken are after going through the classification stages of DBN. Artificial features used in the form of likelihood value of a sample of all sleep stages. So the number of features obtained by the number of sleep stages where the value of features is probability sample. Therefore,

978-1-5386-0549-3/17/$31.00 ©2017 IEEE

this new structure is a contribution of this paper for sleep stage classification. To analysis the influence of this proposed structure, this paper will compare it to the Bi-LSTM.

II. Methods

There are two primary ways in the research, namely Deep Belief Networks (DBN) and Bidirectional Long Short-Term Memory (Bi-LSTM)

A. Deep Belief Networks

Restricted Boltzmann Machine (RBM) is one type of Boltzmann Machine. RBM consisted of some units and divided into the visible layer and hidden layer [7]. Each unit on the visible layers connected to the unit in the hidden layer, but each unit on the same layer has no relationship between them. Also, there is no connection between visible groups or the hidden ones. Deep Belief Networks (DBN) can solve this problem by compiling multiple RBMs, where the hidden layers of an RBM will become visible layers for other RBMs as illustrated in Fig. 1. DBN is a probabilistic generative model that contains multiple layers of hidden variables, where each layer can capture the correlation between the activities of the hidden feature on the layer below [8]. In the DBN, each layer consists of a set of units with a binary or real-valued value. Although DBN has a hierarchical structure with high representative power, it can be easily trained greedy through layer by layer for each RBM.

The training process of DBN has two processes. First, the RBM at the bottom is trained with training data regardless of RBM on it. Second, the above RBM learned by the output of the lower RBM by the unsupervised method.

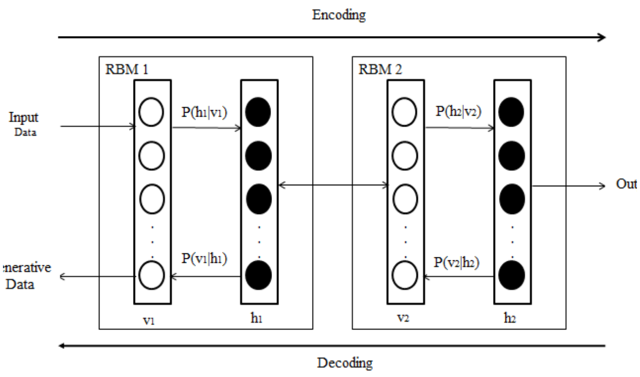

Fig. 1. Architecture of DBN

A generative data obtained through unsupervised pre-trained on the first RBM. The process is done through encoding from the input data to the second RBM. After being handled by the second RBM, the results returned to the first RBM via decoding [9]. Probability when the encoding and decoding process calculated by using equations 1 and 2.

$$P(h_j|v) = \frac{1}{1 + exp^{a_j + \Sigma_i w_{ij} v_i}} \tag{1}$$

$$P(v_i|h) = \frac{1}{1 + exp^{b_i + \Sigma_j w_{ij} h_j}} \tag{2}$$

Where

a and b are bias vector of the visible and hidden layer

w is a weight matrix that represents the connection between visible and hidden layer.

Specifically, in this study, the top layer of DBN was added to be able to classify it by using the supervised method. The final output of DBN in this study is the probability value of each class label for each sample.

B. Bidirectional LSTM

Long Short-Term Memory (LSTM) is one type of Recurrent Neural Networks (RNN). The system consists of several memory blocks where each block contains several gates [10]. The gate, inside the LSTM, is a perceptron and consists of 3 types of gates: input, output, and forget gate. Input and output gate function for setting input and output of the network while forget gate about setting the cell memory. So gates have the important role in LSTM because they have responsibility for setting up memory cells in the processing of storing information. In particular, LSTM is suitable to be applied to processing series data because the system works based on sequence information from data[11]. However, LSTM can only process data in the previous order. Therefore, bi-directional LSTM is applied to be able to process data in two directions i.e. forward and backward in two separate hidden layers. Thus, Bidirectional Long-Short Term Memory (BI-LSTM) consists of two LSTM [12]. To process the data, the first LSTM will read data from left to right while the second LSTM reads data from right to left, as illustrated in Fig. 2.

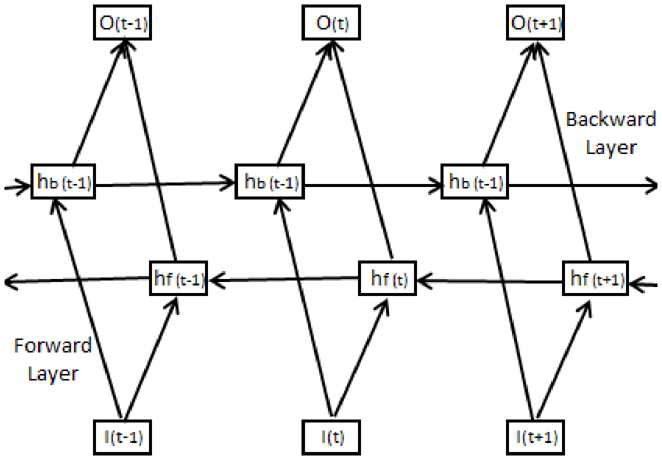

Fig. 2. Architecture of Bi-LSTM

h_t from Bi-LSTM comes from h_t of the forward layer ($h_{f(t)}$) and h_t of the backward layer ($h_{b(t)}$). h_t is calculated using the equation 3.

$$h_t = h_{f(t)} \oplus h_{b(t)} \tag{3}$$

The calculation of forwarding layer based on the equation 4.

$$h_{f(t)} = o_{f(t)} \odot \tanh(c_{f(t)}) \tag{4}$$

$$o_{f(t)} = \sigma(w_{f(o)}x_{f(t)} + v_{f(o)}h_{f(t-1)} + b_{f(o)}) \tag{5}$$

$$c_{f(t)} = z_{f(t)} \odot c_{f(t-1)} + i_{f(t)} \odot \tanh(w_{f(c)}x_{f(t)} + v_{f(c)}h_{f(t-1)} + b_{f(c)}) \tag{6}$$

$$z_{f(t)} = \sigma(w_{f(z)}x_{f(t)} + v_{f(z)}h_{f(t-1)} + b_{f(z)}) \tag{7}$$

$$i_{f(t)} = \sigma(w_{f(i)}x_{f(t)} + v_{f(i)}h_{f(t-1)} + b_{f(i)}) \tag{8}$$

The other hand, calculation of backward layer based on the equation 9.

$$h_{b(t)} = o_{b(t)} \odot \tanh(c_{b(t)}) \tag{4}$$

$$o_{b(t)} = \sigma(w_{b(o)}x_{b(t)} + v_{b(o)}h_{b(t-1)} + b_{b(o)}) \tag{5}$$

$$c_{b(t)} = z_{b(t)} \odot c_{b(t-1)} + i_{b(t)} \odot \tanh(w_{b(c)}x_{b(t)} + v_{b(c)}h_{b(t-1)} + b_{b(c)}) \tag{6}$$

$$z_{b(t)} = \sigma(w_{b(z)}x_{f(t)} + v_{b(z)}h_{b(t-1)} + b_{b(z)}) \tag{7}$$

$$i_{b(t)} = \sigma(w_{b(i)}x_{f(t)} + v_{b(i)}h_{f(t-1)} + b_{b(i)}) \tag{8}$$

Where O is output gate, C is a cell, V is feature vector, W is a weight matrix, σ is the sigmoid function, x is input data, b is a biased matrix, z is a forget gate, and i is input gate.

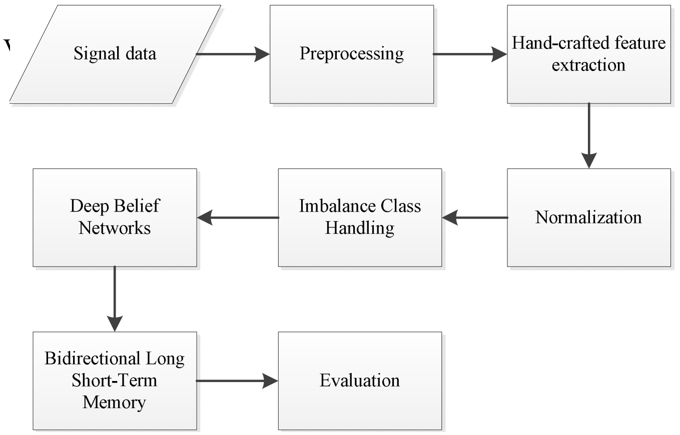

Fig. 3. Flowchart

III. SYSTEM DESIGN

A. Data

Data obtained from St. Vincent's University Hospital and University College Dublin. It can be achieved through https://www.physionet.org/pn3/ucddb/. The data used consisted of 25 patients (21 men and four women) who had slept disorder. The signals used are electroencephalography (EEG) of brain activity with one channel, electromyography (EMG) of skeletal muscle activity with one channel, and two channels of electrooculography (EOG) of eye movements. Sleep stage on each segment is labeled by an expert sleep technologist based on the score of the Rechtschaffen and Kales system.

This study only has five labels of sleep stages: slow wave sleep (SWS), Stage 1, Stage 2, rapid eye movement (REM), and awake. But among labels, they have a different number of sleep segments so that some labels have twice the number of samples.

B. Flowchart

The proposed sleep stage classification is illustrated in Fig. 1. The frequency set for notch filtering is 50 Hz. Furthermore, band-pass filtering assigns filter values of 10-32 Hz on EMG, and 0.3-32 Hz on EEG and EOG. After band-pass filter, down-sampled on data up to 64 Hz. The result of preprocessing data extracted for 28 handcrafted features. Table I describes the detail of feature extraction.

TABLE I. 28 HAND-CRAFTED FEATURES

No	Methods	Signal Type	Descriptions
1	Fast Fourier Transform	EEG, EMG, and EOG	The FFT aims to extract relative power in frequency bands for the signal. The values in frequency bands of Delta, Theta, Alpha, Beta, Gamma for each signal as becoming features for the classification
4	Median of the absolute value	EMG	a feature that appreciated the median value of EMG
5	Eye correlation coefficient	EOG	a function that represents a value of eye correlation between left and right eye
6	Kurtosis	EEG, EMG, and EOG	The kurtosis is considered a form of distortion from the standard curve. It measured by comparing the tangled shape of the data distribution curve with the specifications curve.
9	Standard Deviation	EOG	Standard deviation is used to determine how the distribution of data in the sample in sleep data, and also to know how close the sample to the average.
10	Entropy	EEG, EMG, and EOG	Entropy is used to measure the irregularity or complexity of the signal data
13	Spectral mean	EEG, EMG, and EOG	The average value of spectral signals is used as a feature to see the difference between each sleep stages to the average so that the difference between each sleep stages can be measured.
16	Fractal Exponent	EEG	The fractal exponent of EEG associated with the success of early resuscitation in sleep apnea patient

The results of feature extraction are each segment has 28 elements from three signals used. The next 28 features are normalized. The normalization process is done so that the difference in scale or range of each recording of each feature does not cause errors in model formation during the training process of the model.

Due to differences in the number among sleep labels/classes are significant then the system also handles this imbalance labels by eliminating the number of major classes based on the minor classes, but the sequence information among labels still kept. The process of elimination is done selecting samples from the primary class at random.

DBN implemented by using two layers where each layer consists of 200 hidden units. Both layers trained unsupervised with 300 iterations. The training results in both layers become input for Softmax classifier. This classifier is trained with 50 iterations to obtain probability values for each class in each segment. The output of this DBN generates data with each samples having five features as a representation of probability values for each class.

The Bi-LSTM mechanism applied to the input is derived from the DBN. The length of data sequences in Bi-LSTM, and DBN Bi-LSTM in this study is 15. It means that the processing of x_t (data at time t) based on x_{t-14} to x_t data for the forward direction, and x_{t+14} to x_t for the reverse direction. Filter size (dimensionality for the output space) used is 64. The activation function of Softmax classifier with optimizer is Root Mean Square Propagation. The training process is done as many as 100 epoch. In this study, DBN implementation is based on Martin Langkvist's research [4], while Bi-LSTM based on Keras which can be downloaded from https://github.com/fchollet/keras.

IV. RESULTS

To see the performance of proposed method, then the test is done based on 25 leave-one-out cross validation. Furthermore, this result also compared DBN Bi-LSTM with others methods, namely DBN, Bi-LSTM, and also DBN HMM (Hidden Markov Models).

TABLE II. RESULTS OF THE SLEEP STAGE CLASSIFICATION

%	DBN	DBN HMM	Bi-LSTM	DBN Bi-LSTM
Precision	51.53	64.22	84.92	86.12
Recall	48.43	62.63	71.41	72.39
F-Measure	53.82	69.05	74.32	75.94
Training Time (s)	3213.9	3213.98	5131.57	5742.55

The average performance of classification can see in Table II. The lowest performance obtained by DBN with precision, recall, F-measure value of 51.53%, 48.43%, and 53.82%. It indicates that the non-sequence classifier is not good to apply to the sleep stage classification. Data sequence information is an essential factor to consider in modeling on sleep stage classification. But of course, since it does not involve using sequence data as modeling information then DBN has the smallest training time compared to other classifiers.

TABLE III. DBN

%	Classified as				
	SWS	*Stage 2*	*Stage 1*	*REM*	*Awake*
SWS	79.15	14.05	3.63	2.57	0.59
Stage 2	20.98	51.68	11.22	13.50	2.61
Stage 1	6.27	27.30	26.27	10.04	30.13
REM	4.68	16.27	7.74	63.57	7.74
Awake	1.92	4.54	19.14	13.08	61.32

TABLE IV. DBN HMM

%	Classified as				
	SWS	*Stage 2*	*Stage 1*	*REM*	*Awake*
SWS	90.24	9.57	0.02	0.01	0.16
Stage 2	4.14	82.86	6.92	5.71	0.37
Stage 1	0.14	30.39	37.24	5.65	26.58
REM	0.00	11.02	2.99	80.59	5.40
Awake	0.01	3.06	13.66	7.70	75.58

TABLE V. BI-LSTM

%	Classified as				
	SWS	*Stage 2*	*Stage 1*	*REM*	*Awake*
SWS	69.45	0.62	3.16	15.81	10.97
Stage 2	0.12	89.81	9.15	0.26	0.66
Stage 1	0.83	6.77	75.02	10.79	6.60
REM	29.75	2.17	24.12	32.94	11.03
Awake	6.38	0.75	11.96	6.26	74.65

In the classification using DBN HMM, DBN acts as quantization for inputs from HMM. HMM only accepts one-dimensional data so that quantization becomes a mandatory step to convert data into one dimension. The implementation of HMM in this study used five states for HMM modeling, as Martin Langkvist et al. implemented for the same dataset [4]. The results obtained that the classification performance increased above 10%. It reinforces the fact that the use of information on the sequence of data becomes an important factor in the sleep stage classification, as HMM has done. But both precision, recall, and F-measure it has still below 70%. The processing time for HMM is fast because the processed data is only one dimension. Of course, because in this study HMM also applies DBN then the processing time becomes longer.

At a different point of view, Bi-LSTM is a sequence classifier but can apply data of more than one dimension.

Based on Table II, Bi-LSTM performs better than DBN HMM. Precision from Bi-LSTM has a difference of about 20% above DBN HMM. Recall and F-measure Bi-LSTM above 70%. However, due to the use of 28 handcrafted features, Bi-LSTM has a training time of 5131.57 seconds for modeling.

Furthermore, in the fourth test, Bi-LSTM implemented DBN as an artificial feature extraction. Based on the tests performed, DBN Bi-LSTM performance is best compared to all tested classifier. It shows the artificial features of DBN in the form of probability values from the sample of all sleep stages can be optimal artificial features for Bi-LSTM. Another advantage of DBN is to reduce the amount of training time during modeling at the Bi-LSTM stage. It is because the modeling uses only five artificial features. However, due to DBN, the overall training time of DBN Bi-LSTM has the greatest training time.

The performance of the four methods in detail is explained in the confusion matrix of Tables III, IV, V, and VI. In Table III, DBN most easily classifies SWS while most difficult in the classification of Stage 1. Stage 1 is more classified as SWS. The same goes for DBN HMM in Table IV. But stage 1 is the least classified as SWS.

In Table V, the SWS classified as SWS 69.45%, REM 15.81%, and 10.97%. The class of stage 1, stage 2, and awake are well classified by the system because they successfully classified as much as 89.81%, 75.02%, and 74.65%. But the system difficult to classify REM correctly because it only reached 32.94%. The rest of this label classified as a label of SWS 29.75% and Stage 1 24.12%. The causes of system failure are that REM has no significant difference compared to other stages if it only uses 28 features, or this label has a high similarity with other labels.

In Table VI, by combining it with DBN, the system is better than to classify the sleep stages. SWS label is precise to classify as SWS, which was up 69.45% to 70.37% and awake was classified exactly as awake from 74.65 to 75.56%. In particular, for the REM label, there was a significant increase of 4.28% if it compared without Bi-LSTM. As same as Table V, REM is still hard to classify. This label also classified as SWS and Stage 1 for many samples.

TABLE VI. DBN BI-LSTM

%	Classified as				
	SWS	*Stage 2*	*Stage 1*	*REM*	*Awake*
SWS	70.37	0.25	3.14	17.47	8.77
Stage 2	0.24	89.36	10.01	0.24	0.15
Stage 1	0.92	6.03	75.63	10.83	6.58
REM	28.37	0.83	25.52	37.22	8.06
Awake	6.54	0.32	12.54	5.05	75.56

V. CONCLUSION

Based on Table II, it can be concluded several things, namely:

1. The use of nonsequence classifier like DBN has the lowest performance. It is because this method ignores the data sequence factors in model formation in each class.
2. DBN HMM as sequence classifier can improve classification performance above 10% when compared with DBN. However, precision, recall, and F-measure are obtained only below 70%. It is a consequence of processing data to t only based on x_{t-1} or x_{t+1} so that the range of neighboring t-data involved is minimal.
3. Bi-LSTM is better than DBN HMM because the analysis of t data based on an extensive range. In this study, the length of the sequences used is 15 so that the size of the range is made from x_{t-14} to x_{t+14}
4. Implementation of DBN as an artificial feature extraction proved to be effective in improving classification performance so that DBN Bi-LSTM has the best performance compared to other methods in this research.
5. Training time DBN lowest while DBN Bi-LSTM highest.

Tables III, IV, V, and VI also show that REM is the most classified sleep stage.

ACKNOWLEDGMENT

The Author thanks to the Indonesian Endowment Fund for Education (LPDP) and Machine Learning and Computer Vision Laboratory, Universitas Indonesia that contributed and supported the study.

This work also supported by Center of Excellence for Higher Education Research Grant funded by Indonesian Ministry of Research and Higher Education. This paper also backed by GPU grant from NVIDIA.

REFERENCES

[1] L. Besacier, et al. "Automatic speech recognition for under-resourced languages: A survey." Speech Communication 56 (2014): 85-100.

[2] K. Mozafari, N. M. Charkari, H. S. Boroujeni, and M. Behrouzifar, “A Novel Fuzzy HMM Approach for Human Action,” Knowl. Technol., vol. 295, pp. 184–193, 2012.

[3] I. N. Yulita and H. L. The, “Fuzzy Hidden Markov Models for Indonesian Speech Classification,” J. Adv. Comput. Intell. Intell. Informatics, vol. 16, no. 3, pp. 381–387, 2012.

[4] M. Langkvist, L. Karlsson, and A. Loutfi, “Sleep Stage Classification Using Unsupervised Feature Learning,” Adv. Artif. Neural Syst., pp. 1–9, 2012.

[5] L. Brocki and K. Marasek, “Deep Belief Neural Networks and Bidirectional Long-Short Term Memory Hybrid for Speech Recognition,” Arh. Acoust., vol. 40, no. 2, pp. 191–195, 2015.

[6] R. Brueckner and B. Schuller, “Social Signal Classification Using DEEP BI-LSTM Recurrent Neural Networks,” in IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP), 2014, pp. 4856–4860.

[7] P. Tamilselvan, and P. Wang. "Failure diagnosis using deep belief learning based health state classification." Reliability Engineering & System Safety 115 (2013): 124-135.

[8] R. Salakhutdinov. "Learning deep generative models." Annual Review of Statistics and Its Application 2 (2015): 361-385.

[9] V. Singhal, A. Gogna, and A. Majumdar. "Deep Dictionary Learning vs. Deep Belief Network vs. Stacked Autoencoder: An Empirical Analysis." International conference on neural information processing. Springer International Publishing, 2016.

[10] H. Sak, A. Senior, and F. Beaufays. "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." Fifteenth Annual Conference of the International Speech Communication Association. 2014.

[11] I. Sutskever, O. Vinyals, and Q. V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.

[12] R. Zhao, R. Yan, J. Wang, and K. Mao, “Learning to Monitor Machine Health with Convolutional Bi-Directional LSTM Networks,” Sensors, vol. 17, no. 273, pp. 1–18, 2017.