VNU Journal of Science: Comp. Science & Com. Eng., Vol. 33, No. 2 (2017) 1-13

Deep Learning for Epileptic Spike Detection

Le Thanh Xuyen[1], Le Trung Thanh[2], Dinh Van Viet[2],
Tran Quoc Long[2,*], Nguyen Linh Trung[2], Nguyen Duc Thuan[1]

[1]*Hanoi University of Science and Technology*
[2]*VNU University of Engineering and Technology, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam*

Abstract

In the clinical diagnosis of epilepsy using electroencephalogram (EEG) data, an accurate automatic epileptic spikes detection system is highly useful and meaningful in that the conventional manual process is not only very tedious and time-consuming, but also subjective since it depends on the knowledge and experience of the doctors. In this paper, motivated by significant advantages and lots of achieved successes of deep learning in data mining, we apply Deep Belief Network (DBN), which is one of the breakthrough models laid the foundation for deep learning, to detect epileptic spikes in EEG data. It is really useful in practice because the promising quality evaluation of the spike detection system is higher than 90%. In particular, to construct the accurate detection model for non-spikes and spikes, a new set of detailed features of epileptic spikes is proposed that gives a good description of spikes. These features were then fed to the DBN which is modified from a generative model into a discriminative model to aim at classification accuracy. A performance comparison between using the DBN and other learning models including DAE, ANN, KNN and SVM was provided via numerical study by simulation. Accordingly, the sensitivity and specificity obtained by using the kind of deep learning model are higher than others. The experiment results indicate that it is possible to use deep learning models for epileptic spike detection with very high performance.

Received 24 Jan 2017; Revised 28 Dec 2017; Accepted 31 Dec 2017

Keywords: Electroencephalogram (EEG), Epileptic spikes, Deep Belief Network (DBN), Deep learning.

1. Introduction

Epilepsy is a chronic disorder of the nervous system in the brain. It is characterized by epileptic seizures, which are abnormal excessive discharges of nerve cells. Generally, people with epilepsy may have uncontrollable movement, loss of consciousness and temporary confusion. According to the Epilepsy Foundation and the World Health Organization [45, 46], there are now approximately 65 million people diagnosed with epilepsy and 2.4 million people detected with signs of epilepsy each year in the world. This makes epilepsy the fourth most common neurological disease globally. In developed countries, the number of new cases is between 30 and 50 per 100,000 people in the general population. In developing countries, the figures are nearly twice as high as in the developed

* Corresponding author. E-mail.: tqlong@vnu.edu.vn
https://doi.org/10.25073/2588-1086/vnucsce.156

countries. The figures are remarkable and can increase significantly in the future.

Medical tests are highly important in the diagnosis of epilepsy, including blood-related tests, and brain-related tests using devices such as Electroencephalography (EEG), magnetic resonance imaging (MRI), Computed Tomography (CT). Scalp EEG is used to record and monitor electrical activities of the brain by measuring voltage fluctuations resulting from ionic current flows within the neurons of the brain. The measurement is done by using sensors (electrodes) attached to the skin of the head, receiving electrical impulses of the brain and sending them to a computer. The electrical impulses in an EEG recording is normally characterized by wavy lines with peaks and valleys. Scalp EEG remains the most commonly used medical test for epilepsy, because it is cost-effective and it provides EEG signals with very high temporal resolution required for reading epileptic activity.

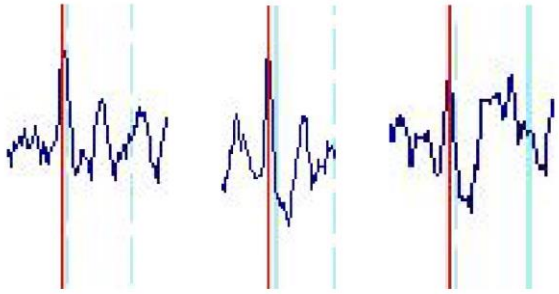

Figure 1. Epileptic spikes in EEG data, marked by the red-lines.

Neurologists usually inspect the EEG recordings on a computer screen and look for signs of epileptic activity, generally called epileptiform discharges, which are abnormal patterns of the brain electrical activity. In this work, we consider one special type of epileptiform discharges, called epileptic spikes, as illustrated in Fig 1. Accurate EEG reading to find spikes greatly depends on the knowledge, experience and skill of the neurologists to avoid misdiagnosis, because various non-epileptic brain activity and artefacts in the recording can look similar to the epileptic spikes. Therefore, it is useful to design automatic EEG software systems that can support the neurologists along, with an automatic spike detection task. Such systems can also save tremendous reading time in 24-hour EEG monitoring. In Vietnam, they can be of even greater support because the lack of skillful neurologists.

Over the last four decades, many methods have been proposed for automatic spike detection, but performance of the existing methods has reached about 90% on average so far. There are two main reasons why the results are still not as good as expected. First, EEG data always contain artefacts due to non-brain activities such as heart beats, eye movements and muscle movements, which are recorded by ECG, EOG and EMG, respectively. Second, the current learning models used in these methods are not good enough, while the epileptic spikes usually have complicated features. In particular, while some spike detection methods are introduced based on simple comparison/filter thresholds between true spikes and possible spikes, such as in [13, 30, 10, 12, 8], some others follow a systematic approach, aimed at revealing different types of hidden information in EEG data, by dividing the automatic detection system into subsystems, performing pre-processing, feature extraction, classification, etc. Often, a spike detection system provides good results if it allows us to exploit the advantages of different algorithms targeting different types of information in the EEG data. Several learning models have been used successfully, such as Artificial Neural Networks (ANN) [42, 27, 20, 28, 23, 37, 36, 5], K-means [35], and Support Vector Machines (SVM) [1].

Recently, deep learning has been attracting a great attention in machine learning. Deep learning exploits various *deep architectures* and specialized learning algorithms to capture multi-level representation and abstraction of data. These deep architectures have achieved several successes and occasionally breakthrough in many applications such as natural language processing, speech recognition, speech synthesis, image processing and computer vision. In particular, recent EEG studies have used deep learning to some extent. For example, Convolutional Neuron Network (CNN) is the first deep learning model applied for EEG seizure prediction [26]. [43, 22, 44] use another

deep learning model called Deep Belief Network (DBN) on EEG data to investigate anomalies related to epilepsy, different sleep states, critical frequency bands for EEG based emotion recognition respectively; other deep learning models are used to explore complicated tasks such as discovery of brain structure [31]; learning brain waves' characteristics [38] using three deep models including CNN, deep learning using linear Support Vector Machine (DL-SVM) and Convolutional Auto Encoders (CAE); classification of EEG data using multichannel DBN [3]. At the same time, [18] applies CNN model to detect epileptic spikes in EEG data. However, since EEG signals are non-stationary and they can vary greatly from patient to patient, there might be not sufficient data (i.e. only 5 patients) to evaluate the performance of the detection system. Furthermore, a performance comparison between CNN and simple shallow learning models as KNN, RF, SVM is provided but the results show insignificant difference.

At the same time, deep learning could be categorized into different classes based on kinds of factors such as architectures, purposes and learning types [9]. Recently, CNNs are well known as the most famous type of deep learning. They are highly effective and commonly used in computer vision, image recognition, and speech recognition with very good results. To our best knowledge, types of CNN, however, may reach their saturation point. If improving, there is just a little bit. So what's next for deep learning? Deep generative models can be the good alternative solutions due to the fact that they are not only directly related to learning theory compared with the inference process of our brain, but also able to go deeper. There are now many types of deep generative models such as Deep Boltzmann Machines [33], Deep Auto Encoders [21, 6], Deep Belief Networks [14] and Generative Adversarial Nets [11]. This motivates us apply the kinds of learning model first.

The studies mentioned above encourage us to find and experiment an improved deep learning model to detect epileptic spikes, as described shortly after. The contributions of this work are: first, we define a detailed feature extraction model for EEG data that is suitable for applying deep learning models; and second, we introduce a systematic approach to apply DBN for epileptic spikes detection.

The paper is organized as follows: In Section 2, we introduce information related directly to our feature extraction and DBN model for classification. Implementation of our methods for detecting spikes is presented in Section 3 and then Section 4 concludes the study with some notes and future works.

2. Methods

2.1. Feature extraction

For large and noisy datasets, feature extraction is a vital preprocessing step. If carried out successfully, feature extraction could reduce the undesired effect of noise and high dimensionality, the main culprits that hinder high performance detection system for EEG data in particular. In this work, multiple methods have been proposed based on the parameters of a spike in time-frequency domain, for example, eigenvector methods [41], spike models with wave features [24], [23] and time-varying frequency analysis [32]. These methods are combined to find a set of measurements characterizing the spikes.

Over a last decade, wavelet transform is valuable in processing non-stationary signals analysis like EEG recordings. In particular, wavelet decomposes the signal $x(t)$ into other signals by varying the wavelet scale a and shift b, which provides different views of the signal and visualizes the signal features. Wavelet transform has been successfully applied in recent studies in EEG such as spike detection and sorting [32]. More specifically, wavelet features of a spike are obtained immediately from the waveforms of the transformed signal, leading to the selection of wavelet scale to be used as input for spike detection systems. The wavelet scale is selected such that the corresponding transformed signal of an epileptic spike is likely to be waveform of the true spike, while wavelet transform of non-spike is disabled. For example, in the recently proposed multi-stage automatic epileptic spike detection system in [5], the authors choose the continuous wavelet transform

(CWT) at 5 scales (from 4^{th} to 8^{th}) that could improve detection performance. In a nutshell, using waveform features of wavelet as input of the classifier could be effective.

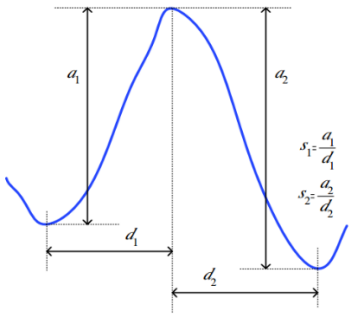

Figure 2. Features of a spike.

Motivated by results from the previously proposed methods and significant advantages of wavelet transform, we introduce a model to extract a set of detailed features for each peaks in EEG data. Seven wavelet features of spikes are obtained from [23] and divided into 4 groups: duration, amplitude, slope and area, shown as in Fig 2. In addition, by enlarging the scale range compared to that of [5], we increase the dimension of input space providing more information about spikes. In particular, the EEG bandwidth is divided into 4 sub-bands including Theta (3.5 - 7.5 Hz), Alpha (7.5 - 12.5 Hz), Beta1 (12.5 - 30 Hz) and Beta2 (30 - 50 Hz) and each sub-band gets 10 scales to obtain total 280 parameter of features in total. These parameters are then fed to the DBN classifier as discussed in the next section.

2.2. Deep belief network for classification

Deep Belief Network (DBN), proposed by Hinton *et al.* [14], is considered as one of the most breakthrough models constructing the foundation for deep learning. DBN consists of two types of neural layers: Belief Network and Restricted Boltzmann Machine, shown as in Fig. 3.

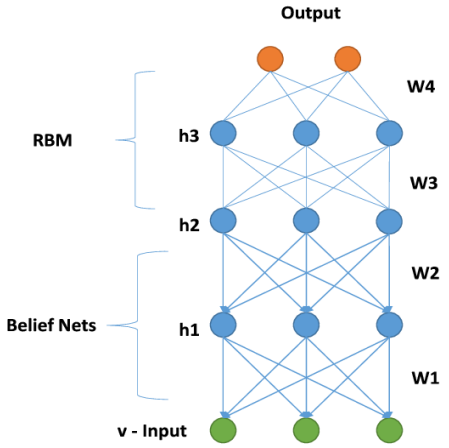

Figure 3. A typical DBN contains 2 Belief Nets and 2 RBMs.

Belief Network

Belief network, or alternately Bayesian network, is often used to contruct the first stages or layers of a DBN, shown as in Fig 3. The network is a causal model which present the cause-effect relationship between input and output layer via Bayesian probability theory [7]. In particular, a belief network connecting two layers using a weighted matrix **W** and the probability of input neurons becoming 1 is as follows

$$P(\mathbf{h}_1(j) = 1) = \frac{1}{1 + e^{-\sum_i \mathbf{h}_2(\mathrm{i})\mathbf{W}_{i,j}}}. \quad (1)$$

One could use this model to infer the state of unobserved units and, in model training, one could adjust the weights to capture the distribution of observed data. Belief network is often trained using many iterations of Markov Chain Monte Carlo (MCMC) which could be very time-consuming. Furthermore, when stacked in a multi-layer network, its inference becomes infeasible due to large number of possible configurations and that convergence is not guaranteed. To circumvent these drawbacks, Hinton *et al.* proposed that one could restrict the connectivity between layers and train the network one layer at a time using a simplified cost function called Contrastive Divergence (CD). This breakthrough [15, 16] will be discussed in the next section.

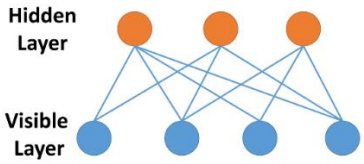

Figure 4. Restricted Boltzmann Machine.

Restricted Boltzmann Machine

Restricted Boltzmann Machine (RBM), a special type of Markov random field, is a simplified Boltzmann Machine. RBM is first introduced in the 1980s [2]. The network consists of two layers: *visible layer* where states (neurons) are observed, and *hidden layer* where the features are detected. RBM only has inter-layer connections and does not allow intra-layer connections [34]. The structure of a RBM is depicted in Fig. 4.

The RBM network simulates the law of thermodynamics in which each state (configurations) of the network is characterized by a energy, given by:

$$E(\boldsymbol{v},\boldsymbol{h}) = -\sum_{i,j} \boldsymbol{v}_i \boldsymbol{h}_j \mathbf{W}_{i,j} - \sum_i a_i \boldsymbol{v}_i - \sum_j b_j \boldsymbol{h}_j. \quad (2)$$

The joint probability over hidden and visible units in a configuration is then defined in terms of energy function:

$$P(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z} e^{-E(\boldsymbol{v},\boldsymbol{h})}. \quad (3)$$

where Z is the *partition function*, i.e. the total energy of all configurations of the network

$$Z = \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}. \quad (4)$$

The probability that the network assigns to a certain visible input vector v is

$$P(\boldsymbol{v}) = \frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}. \quad (5)$$

Given a training set of N input (visible) vectors $v^{(\ell)}, \ell = 1,\ldots,N$, the selection of the model parameters (i.e. the $W_{i,j}, a_i, b_j$'s) follows the Maximum Likelihood Estimation (MLE) principle. The MLE principle states that the best set of parameters should maximize the training data likelihood (or log-likelihood), which is defined as the probability of the training data given a set of parameters. In particular, for RBM, one has to maximize the log-likelihood of $v^{(\ell)}, \ell = 1,\ldots,N$:

$$\max_{\mathbf{w}_{i,j},a_i,b_j} \frac{1}{N}\sum_{\ell=1}^{N} \log \sum_{\boldsymbol{h}} P\left(\boldsymbol{v}^{(\ell)},\boldsymbol{h}\right), \quad (6)$$

where N is the number of training data. One could solve (5) using the gradient methods meaning that one need to compute its derivatives

$$\frac{\partial \log P(\boldsymbol{v})}{\partial \mathbf{W}_{i,j}} = \langle \boldsymbol{v}_i \boldsymbol{h}_j \rangle_{data} - \langle \boldsymbol{v}_i \boldsymbol{h}_j \rangle_{model}, \quad (7)$$

where $\langle . \rangle_{data}$ and $\langle . \rangle_{model}$ are the expectation operators under data and model distributions, respectively. The parameter is then adjusted as

$$\Delta \mathbf{W}_{i,j} = \varepsilon.\left(\langle \boldsymbol{v}_i \boldsymbol{h}_j \rangle_{data} - \langle \boldsymbol{v}_i \boldsymbol{h}_j \rangle_{model}\right), \quad (8)$$

$$\Delta a_i = \varepsilon.\left(\langle \boldsymbol{v}_i \rangle_{data} - \langle \boldsymbol{v}_i \rangle_{model}\right), \quad (9)$$

$$\Delta b_j = \varepsilon.\left(\langle \boldsymbol{h}_j \rangle_{data} - \langle \boldsymbol{h}_j \rangle_{model}\right), \quad (10)$$

with ε is the learning rate.

To compute $\langle . \rangle_{data}$, the expectation under data distribution, one could exploit the fact that there are no direction connections between hidden units in a RBM. This allow one to easily generate an unbiased sample of the state of hidden units via the conditional probability

$$P\left(\boldsymbol{h}_j = 1 \middle| \boldsymbol{v}\right) = \frac{1}{1 + \exp\left(-b_j - \sum_i \boldsymbol{v}_i \mathbf{W}_{i,j}\right)}. \quad (11)$$

Similarly, one could generate an unbiased sample of the state of a visible unit given a hidden vector because there are no connections between units in visible layer, either.

$$P(\boldsymbol{v}_i = 1 | \boldsymbol{h}) = \frac{1}{1 + \exp\left(-a_j - \sum_j \boldsymbol{h}_j \mathbf{W}_{i,j}\right)}. \quad (12)$$

Obtaining the expectation under model distribution $\langle v_i h_j \rangle_\infty$, however, is much more difficult. Generally, one could perform alternative Gibbs sampling for a huge number of iterations starting from a random state of the

visible units, as described in the MCMC algorithm [4]. This is infeasible when the number of units is increasing and later, when RBM layers are stacked in a deep architecture.

Fortunately, the Contrastive Divergence (CD) algorithm [15, 16] can be used to fasten the learning for an RBM. The general idea is to sample all the hidden units in parallel starting from visible units (input), then reconstruct visible units from the sampled hidden units, and finally sample the hidden units once again. The intuition behind this is that after a few iterations the data will be transformed from the target distribution (i.e. that of the training data) towards the model distribution, and therefore this gives an idea in which direction the proposed distribution should move to better model the training data. Empirically, Hinton has found that even 1 cycle of MCMC is sufficient for the algorithm to converge to the acceptable answer. The learning rule is

$$\Delta \mathbf{W}_{i,j} = \varepsilon.\left(\langle \boldsymbol{v}_i \boldsymbol{h}_j \rangle_{data} - \langle \boldsymbol{v}_i \boldsymbol{h}_j \rangle_1\right), \quad (13)$$

$$\Delta a_i = \varepsilon.\left(\langle \boldsymbol{v}_i \rangle_{data} - \langle \boldsymbol{v}_i \rangle_1\right), \quad (14)$$

$$\Delta b_j = \varepsilon.\left(\langle \boldsymbol{h}_j \rangle_{data} - \langle \boldsymbol{h}_j \rangle_1\right), \quad (15)$$

where $\langle \cdot \rangle_1$ represents the expectation operator given by 1 cycle of MCMC. The CD algorithm with 1 cycle (CD_1) is summarized as follows:

- Initialize $\boldsymbol{v}_0$ from input data;
- Sample $\boldsymbol{h}_0 \coloneqq p(\boldsymbol{h}|\boldsymbol{v}_0)$;
- Sample $\boldsymbol{v}_1 \coloneqq p(\boldsymbol{v}|\boldsymbol{h}_0)$;
- Sample $\boldsymbol{h}_1 \coloneqq p(\boldsymbol{h}|\boldsymbol{v}_1)$.

The algorithm described above represents a breakthrough in learning a single layer of Deep Belief Networks (DBN). Several RBM layers could be stacked and configured (i.e. learned) sequentially to obtain multi-level representation of the data. The idea is to used output of previous layers as training data of subsequent layers and one could learn multiple layers at ease. In the next section, we will discuss our method to adapt DBN, a powerful generative model, to use in classification tasks.

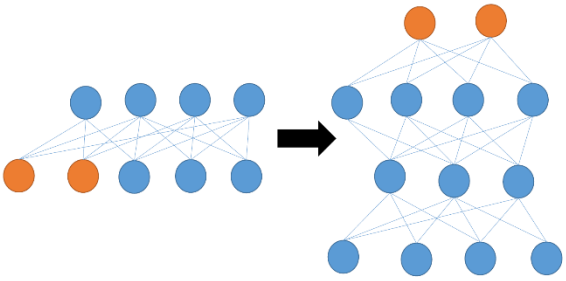

Figure 5. Generative DBN to discriminative DBN.

Deep Belief Networks for EEG Classification

Deep Belief Networks could learn pattern in data even when no labeled sample is available. DBN efficiently models the generative distribution of input data. However, when used in classification tasks such as EEG classification, one needs to augment the architecture of DBN for classification accuracy.

To carry out classification, we add a *discriminative* objective function on top of the existing DBN. There are several possible methods for classification. Firstly, one can use standard discriminative methods which use features (outputs) generated by DBNs as inputs, for example, k-Mean, kNN, logistics regression, SVM [39]. However, a more natural way to add classification capability to DBNs is to directly modify the *generative* DBN model into a *discriminative* DBN model [17]. This method transforms two units of the last RBM into a new stage as shown in Fig 5. To be more specific, we train RBM on each class (we have only two groups: epileptic-spike and non-spike), and then obtain the free-energy of a test data vector for each class. The free energy of a visible vector ($F(\boldsymbol{v})$) is defined as the energy a configuration need to obtain in order to have same probability as all configuration that contain $\boldsymbol{v}$ [17].

For each class-specific RBM, we have that

$$e^{-F(v)} = \sum_h e^{-E(\boldsymbol{v},\boldsymbol{h})} \quad (16)$$

$$F(\boldsymbol{v}) = -\sum_i a_i \boldsymbol{v}_i \sum_j p_j x + \sum_i p_i \log p_i + (1-p_i)\log(1-p_i). \quad (17)$$

It is also calculated by

$$F(\boldsymbol{v}) = -\sum_i a_i \boldsymbol{v}_i - \sum_j \log(1 + e^{x_j}). \quad (18)$$

where $x_i = b_i + \sum_i v_i W_{ij}$ is the total input to hidden unit j, $p_j = \sigma(x_j)$ is the probability that $\boldsymbol{h}_j = 1$ give $\boldsymbol{v}$.

Recall that there are only 2 classes in EEG data, so it is easy to predict the probability of assigning a vector to one class via its free energies as

$$P(class = c \mid \boldsymbol{t}) = \frac{e^{-F_c(\boldsymbol{t})}}{\sum_{d=1}^{2} e^{-F_d(\boldsymbol{t})}}. \quad (19)$$

where $F_t(\boldsymbol{t})$ is a free energy of the test vector $\boldsymbol{t}$ on class c.

3. Experiments

3.1. EEG dataset

The EEG data used in this study are recorded at Signal and Systems Laboratory, University of Engineering and Technology, Vietnam National University using the international standard 10-20 system with 32 channels and representing in EEG with the sampling rate of 256 Hz. Measurements were carried out on 19 patients aged from 6 to 18 years who were detected signs of the epilepsy.

In data collection, we first gather locations of epileptic spikes which are validated by a neurologist, then take 56 data points around each peak position into a segment presenting a spike. After that, 1491 epileptic spike segments (vectors) are combined together into the first class namely "spike". Similarly, we take random peak segments samples from the EEG dataset to create the non-spike class. They are therefore randomly divided into three subsets based on cross validation method: a training and a validation set are obtained from a number of patients; while the remaining patients are used to tested. In a nutshell, we get totally several cases for experiments to measure how good the DBN is.

There is a significant difference in EEG data usage between our implementation and previous method. In the following experiments, we use the raw EEG data instead of filtering out the "noise". In general, the EEG data always consist of many artifacts as mentioned in section 1. This artifacts often lead to difficulty in reliably detectiing epileptic spike. Thus, in previous methods, preprocessing step is highly important to minimize the effect of the noise on the performance of spike detector. In fact, to the best of our knowledge, there has been no study of high performance spike detector in EEG using only raw data. In this work, that features are extracted from unprocessed data using DBN without any filtering also helps the whole detection system performs faster.

3.2. Evaluation metrics

There are various criteria used to measure the performance of a detection system depending on specific fields. In this work, *sensitivity*, *selectivity*, *specificity* and *accuracy*, which are typical statistical measures in machine learning and computer science, are first used to evaluate the quality of our spike detection system. In particular, let's consider that TP and FP are a number of correctly and incorrectly identified epileptic spikes in EEG data respectively; TN, FN are the number of correctly and incorrectly rejected non-spikes, respectively. Therefore, the *sensitivity* (SEN) measures a proportion of correct classification , that is given by

$$\mathrm{SEN} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}; \quad (20)$$

the *selectivity* (SEL) indicates a percentage of spikes that are correctly detected over total spikes detected by the classifier

$$\mathrm{SEL} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}; \quad (21)$$

the *specificity* (SPE) is quite similar to *selectivity* (SEL) but for negative cases

$$\mathrm{SPE} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}; \quad (22)$$

meanwhile the *negative predictive value* (NPV) is a proportion of non spikes identified correctly

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}}; \quad (23)$$

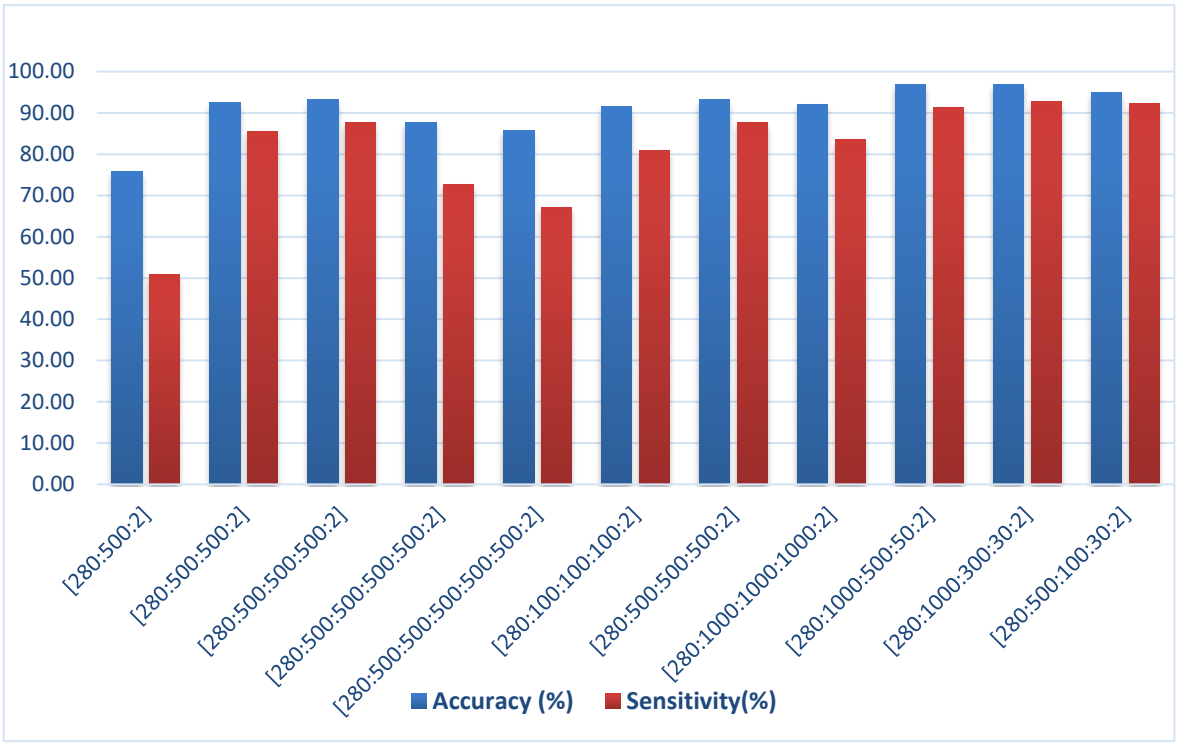

Fig 6. Configurations of the DBN

The *accuracy* (ACC) show hows the classifier makes the correct prediction,

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}}. \qquad (24)$$

The following confusion matrix is another way to illustrate the above evaluation metrics. The performance criteria above are represented as columns and rows of this matrix, as shown in Tab 1.

Table 1. Matrix Confusion

TN	FN	NPV
FP	TP	SEL
SPE	SEN	ACC

Finally, we also use Receiver Operator Characteristic (ROC) curve to visualize the performance of the system. The curve is drawn by plotting **true positive rate** based on SEN and **false positive rate** that can be calculated as $1 -$ SPE. ROC analysis allows us get a trade offs between benefits and costs to make a decision.

3.3. Results

Our experiments are implemented in MATLAB 2015b on Intel core i7 processor and 8G RAM machine. In the experiments, DBN training is performed through three steps including pre-training of each layer; training all layers and fine-tuning of all with back-propagation. The goal of the training is to learn the weights and biases between each layer and reconstruction so that the network's output are as close to the input as possible. In this section, we would like to estimate how good the DBN implement in practice via three estimation cases: (1) estimating the best DBN's configuration, (2) testing the DBN based on the cross validation method and (3) comparing the DBN with previously proposed methods and the state of the art deep learning methods.

Table 2: Quantitative statistics of the DBN based on the Leave-One-Out Cross Validation method

Patient	Spikes/Non-Spikes	SEN	SPE	Patient	Spikes/Non-Spikes	SEN	SPE
1	8/190	75.00%	97.89%	9	4/380	100%	100%
2	44/190	95.45%	97.37%	10	635/190	97.95%	98.95%
3	22/190	81.82%	99.47%	11	22/190	86.36%	97.89%
4	28/380	85.71%	99.70%	12	5/190	100%	89.47%
5	4/380	50.00%	98.42%	13	1/190	0%	100%
6	351/190	84.90%	95.79%	14	24/190	95.68%	99.47%
7	8/190	100%	98.95%	15	2/190	0%	97.36%
8	21/380	80.95%	100%	16	11/190	81.82%	85.26%

First, several different configurations of the DBN in terms of the number of hidden layers and hidden units are tested to choose the best result. We configure the DBN as following. The number of units in input and output layer corresponds to the true length of vector feature input and possible classifications on EEG data. The number of units in each hidden layers will be tested in simulation to find the best number of hidden units. Besides, we also let the number of hidden layers vary. Those settings of number of layers and number of units constitute several configurations of the DBN. We test these configurations to examine the best deep architecture of DBN for our EEG dataset.

It may be intuitive that if the DBN has many more hidden layers, the network is able to learn more complex features in dat with high accuracy. However, this can be misconception. We first use one hidden layer for training (then the total system contains input layer - a hidden layer - output layer), and the classification accuracy is not good. We then add another hidden layer (with same number of units to the first layer) and get a good result. Again, another hidden layer is put into the DBN that gives a improved result. As far, the more depth is good; hence, we add another layer with encouragement. Suddenly, the result fell down, one more time, we try inserting more layers into the deep network, but it is not encouraging, either.

Table 3. The sample EEG dataset to investigate various configurations of the DBN model for the best result

	Training	Validation	Testing
Epileptic Spike	978	123	390
Non-Spike	2030	377	760
Total	3010	500	1150

In practice, when dealing with the case of a sample dataset as in Tab 3, the typical results are shown statistically in Fig 6. Specifically, 4 first items give the result for varying number of hidden layers and fixed number of hidden units, while the next items gives the results for fixed number of hidden layers and varying number of units in each or every hidden layer. It can be seen that the configuration of [1 input, 3 hidden layers, 1 output] allows us to have the best classification accuracy. Next, the results for the cases of varying number of units confirm that the number of units should be under a threshold for each layer to obtain best results. If they overcome this value, the classification accuracy will drop. This negates the intuition that the more number of neurons in each layer, the more efficient performance. By comparing across training, we observe that we observe that the

DBN's configuration of [1 input, 3 hidden layers, 1 output] with [280:1000:300:30:2] neurons has the highest average performance in item of *sensitivity, selectivity, specificity,* and *accuracy* 92.82%, 97.83% , 96.41%, and 96.87% respectively. In particular, the results are shown statistically in Confusion Matrix in Fig 7. It is clear that 362 epileptic spikes are correctly detected that corresponds to 97.8% and 92.83% of all peaks detected by DBN and the neurologist respectively. Only 8 non-spikes are detected as epileptic spikes and this corresponds to 0.7% of 1150 peaks in the testing data. More specifically, out of 390 true epileptic spikes, 92.83% are correct and 7.2% are wrong. At the same time, total evaluation metrics measuring non-spikes are very well with NPV and SPE be 98.9%, 96.4% respectively. Overall, 96.9% of prediction are correct and 3.1% are wrong detection.

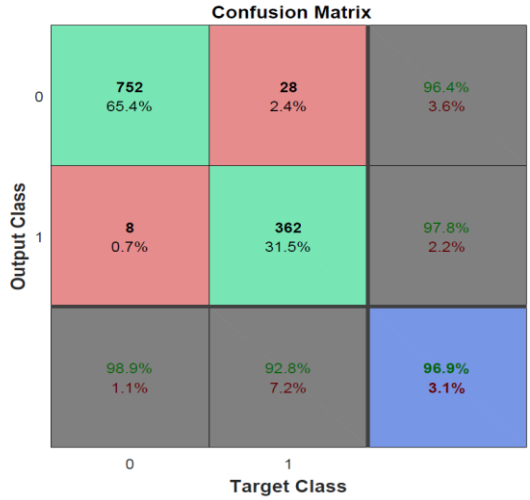

Figure 7. Resulting confusion matrix.

Second, several experiments are implemented on many datasets to estimate the performance of the DBN in practice. Recall that, the EEG signals are nonstationary which vary not only from patient to patient, but also from day to night in each patient. This leads to the fact that results may not be good if the testing patient is greatly different both in terms of the number of epileptic spikes and their characteristic shape from the training patients. At the same time, leave-one-out cross-validation (LOO-CV) is a well-know tool for estimating the performance of classification systems that can provide a conservative evaluation [19]. In this work, the whole EEG dataset composed of 19 patients are randomly split into training, validation and testing sets based on the LOO-CV. In each observation, the best DBN's configuration is fitted using a training data composed of 18 patients and then tested by a remaining patient. The measurement is repeated until the last patient is done.

The experimental results are shown statistically in the Tab 2. It can be clearly that, the estimation of emphspecificity is stable in all tests which is reasonable at 95% to 100% due to the fact that the number of non spikes for testing are large compared with the testing epileptic spike, meanwhile the *sensitivity* seems to be different in patients. Accordingly, among the observations, the patient number 7 and 8 reach the highest *sensitivity* of 100%; whereas the DBN can not detect any epileptic spikes of patient number 13 and 15 leading to the lowest result at 0% or the model returns a *sensitivity* of 50% from patient number 5. It may be caused by the fact that the patients have a few spike which can be considered as anomalies, so it is hard to capture them. In addition, the statistics indicate that the more epileptic spike we obtain from the testing patient, the higher accuracy the DBN can predict at. For examples, 622 spikes of patient number 10 are correctly detected over the total number of 635 spikes with a precision of 97.95%; and in the case of the patient number 14, the experimental results are very high when the percentage of epileptic spikes and non spikes detected correctly is 95.68% and 99.47% respectively. In other cases, the outputs returned from patients with more than 20 spikes are quite good and stable in the range *sensitivity* of 80% to 86%.

Finally, a performance comparison between using the DBN and other learning models was provided via numerical study by simulation. In this work, there are the ANN, deep autoencoder (DAE), support vector machine (SVM) and K-nearest neighbor (kNN). In particular, the ANN is organized by an input layer, two hidden layers and an output layer followed the way of Liu [23] and Dao [5]. The DAE which is a deep

generative model is modified into a discriminative model to be aiming to predict epileptic spikes that is composed of three stages including encoder, decoder and softmax layer [6]. The SVM and kNN, which are well-know models, are already applied to classify epileptic spikes, shape waves and emotion in EEG data in [1, 29] and [25] respectively. All the models are trained and tested on the same above EEG dataset.

The results are show statistically and graphically in Tab 4 and Fig 8. It is clear that all the quality evaluation including *sensitivity*; *specificity* and area undercurve (AUC) of the DBN are better than that of other models. Moreover, using DBN consumes less training time than using others for the reason which the training time of DBN can be reduced by the decreasing the number of iterations to convergence in CD algorithm while SVM, kNN and ANN are very time-consuming in the training process due to the high-dimensional input vector space. Specifically, the SEN, SPE of the DBN classifier are 87.35%, 97.89% respectively and better 20% than the classifier ANN, meanwhile, only 58.64% and 28.40% of true spikes are correctly detected by SVM and kNN. It may be caused by the fact that the EEG dataset used in this work is raw without filtering and removing artifacts. Therefore, using shallow architectures are not useful for this work. Surprisingly, the deep DAE model can not detect any spikes and provide a worthless result with very low AUC of approximately 0.5. It indicates that not all deep learning models are suitable for this problem. In addition, the experiments show that the DBN reaches the biggest AUC of 0.9597 representing an excellent system which providing better performance than other models. Once again, this emphasizes the advantage and efficiency of DBN in epileptic spikes detection.

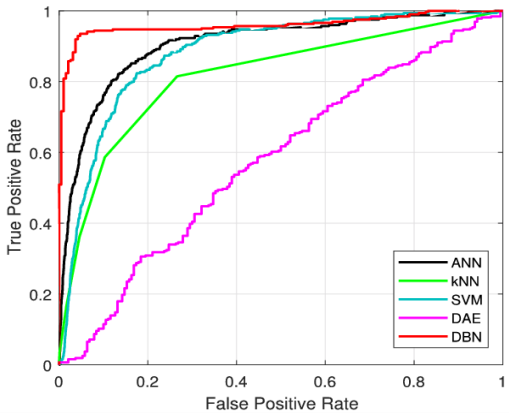

Figure 8. ROC curves for some learning models trained on the EEG data.

Table 4. A performance comparison between the DBN and other learning models

Model	SEN	SPE	AUC
DBN	87.35%	97.89%	0.9597
DAE	0%	100%	0.5232
ANN	65.74%	91.72%	0.8918
SVM	58.64%	92.53%	0.8815
kNN	28.40%	95.42%	0.8058

4. Conclusions

In conclusion, we have applied the DBN model as a classifier to detect epileptic spikes in EEG signal. The training process show that the DBN can learn hidden features in EEG data which distinguish between epileptic spikes and non spikes group with high accuracy. The experiment results not only indicates that learning high-level representations of EEG data can be achieved successfully for spike detection, but also emphasizes the advantage and efficiency of DBN in epileptic spikes detection. In addition, we also compare the performance of the detection system between using the DBN and other learning models like SVM, kNN, ANN, DAE. Accordingly, the results returned by the kind of deep learning models are better than those earlier methods.

In the near future, we would like to build a new model to get more suitable features for deep networks with better classification result. We will also continue to complete the DBN model and try to use the other state of the art deep learning models, adjust the parameters of these networks to determine which is the best model for detecting spikes in EEG signal. Moreover, to get higher quality, we will consider improving preprocessing with more appropriate design of filters, perceptrons to get clearer data before training deep learning models.

Acknowledgments

This work was supported by Project CN.16.07 granted by the University of Engineering and Technology, Vietnam National University, Hanoi. Part of this work was presented in the bachelor graduation thesis of Le Trung Thanh [40]. The EEG data used in this work were part of the EEG epilepsy database constructed within the framework of Project QG.10.40 funded by Vietnam National University Hanoi.

References

[1] N. Acr and C. Güzelis. Automatic spike detection in EEG by a two-stage procedure based on support vector machines. *Computers in Biology and Medicine*, 34(7):561–575, 2004.

[2] D.H. Ackley, G.E. Hinton, and T.J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

[3] A.M. Al-kaysi, A. Al-Ani, and T.W. Boonstra. A multichannel deep belief network for the classification of EEG data. In *International Conference on Neural Information Processing*, pages 38–45, 2015.

[4] C. Andrieu, N.D. Freitas, A. Doucet, and M.I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1-2):5–43, 2003.

[5] N.T.A. Dao, N.L. Trung, N.V. Ly, T.D. Tan, N.T.H. Anh, and B. Boashash. A multistage system for automatic detection of epileptic spikes. REV Journal on Electronics and Communications, 8(1–2):1–13, 2018

[6] P. Baldi. Autoencoders, Unsupervised learning, and deep architectures. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 37–49, 2012.

[7] I. Ben-Gal. Bayesian networks. *Encyclopedia of Statistics in Quality and Reliability*, 2008.

[8] C. F. Boos, M. d. C. V. Pereira, F. I. M. Argoud and F. M. de Azevedo. Morphological descriptors for automatic detection of epileptiform events. In *Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 2435–2438, 2010.

[9] L. Deng, D. Yu, et al. Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4):197–387, 2014.

[10] P.G.D Oliveira, C. Queiroz, and F.L. Da Silva. Spike detection based on a pattern recognition approach using a microcomputer. *Electroencephalography and Clinical Neurophysiology*, 56(1):97–103, 1983.

[11] G. Ian, P.A. Jean, M. Mirza, B. Xu, W.F. David, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[12] J. Gotman and L. Wang. State dependent spike detection: validation. *Electroencephalography and Clinical Neurophysiology*, 83(1):12–18, 1992.

[13] J. Gotman and P Gloor. Automatic recognition and quantification of interictal epileptic activity in the human scalp EEG. *Electroencephalography and Clinical Neurophysiology*, 41(5):513–529, 1976.

[14] G.E. Hinton, S. Osindero, and Y.W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[15] G.E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771-1800, 2006.

[16] G.E. Hinton and M.A. Carreira-Perpinan. On contrastive divergence learning. In *AISTATS*, pages 33–40, 2005.

[17] G.E. Hinton. A practical guide to training restricted Boltzmann machines. *Neural networks: Tricks of the trade*, pages 599-619, 2012.

[18] A. R. Johansen, J. Jin, T. Maszczyk, J. Dauwels, S. S. Cash and M. B. Westover. Epileptiform spike detection via convolutional neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 754–758, 2016.

[19] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*, pages 1137–1143, 1995.

[20] C.W. Ko and H.W. Chung. Automatic spike detection via an artificial neural network using raw

EEG data: Effects of data preparation and implications in the limitations of online recognition. *Clinical Neurophysiology*, 111(3): 477–481, 2000.

[21] A. Krizhevsky and G.E. Hinton. Using very deep autoencoders for content-based image retrieval. In *ESANN*, 2011.

[22] M. Längkvist, L. Karlsson, and A. Loutfi. Sleep stage classification using unsupervised feature learning. *Advances in Artificial Neural Systems*, 2012:5, 2012.

[23] H.S. Liu, T. Zhang, and F.S. Yang. A multistage, multimethod approach for automatic detection and classification of epileptiform EEG. *IEEE Transactions on Biomedical Engineering*, 49(12): 1557–1566, 2002.

[24] Y.C. Liu, C.C.K. Lin, J.J Tsai, and Y.N. Sun. Model-based spike detection of epileptic EEG data. *Sensors*, 13(9):12536–12547, 2013.

[25] R.M. Mehmood and H.J. Lee. Emotion classification of EEG brain signal using SVM and KNN. In *IEEE International Conference on Multimedia & Expo Workshops*, pages 1–5, 2015.

[26] P. Mirowski, D.Madhavan, Y. LeCun, and R. Kuzniecky. Classification of patterns of EEG synchronization for seizure prediction. *Clinical Neurophysiology*, 120(11):1927–1940, 2009.

[27] Ö. Özdamar and T. Kalayci. Detection of spikes with artificial neural networks using raw EEG. *Computers and Biomedical Research*, 31(2): 122–142, 1998.

[28] C. C. C. Pang, A. R. M. Upton, G. Shine, and M. V. Kamath. A comparison of algorithms for detection of spikes in the electroencephalogram. *IEEE Transactions on Biomedical Engineering*, 50(4):521–526, April 2003.

[29] Y. Pan, S. S. Ge, F. R. Tang, and A. A. Mamun. Detection of epileptic spike-wave discharges using SVM. In *IEEE International Conference on Control Applications*, pages 467–472, 2007.

[30] G. Pfurtscheller, G. Fischer. A new approach to spike detection using a combination of inverse and matched filter techniques. *Electroencephalography and Clinical Neurophysiology*, 44(2):243–247, 1978.

[31] S.M. Plis, D.R. Hjelm, R. Salakhutdinov, E.A. Allen, H.J. Bockholt, J.D. Long, H.J. Johnson, J.S. Paulsen, A.T. Jessica, and V.D. Calhoun. Deep learning for neuroimaging: a validation study. *Frontiers in Neuroscience*, 8, 2014.

[32] R.Q. Quiroga, Z. Nadasdy, and B.S. Yoram. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Computation*, 16(8): 1661–1687, 2004.

[33] R. Salakhutdinov and G.E. Hinton. Deep Boltzmann machines. In *Artificial Intelligence and Statistics*, pages 448–455, 2009.

[34] R. Salakhutdinov. Learning deep generative models. *Annual Review of Statistics and Its Application*, 2:361–385, 2015.

[35] T.W. Shen, X. Kuo, and Y.L. Hsin. Ant K-means clustering method on epileptic spike detection. In *International Conference on Natural Computation, v*olume 6, pages 334-338, 2019.

[36] V. Srinivasan, C. Eswaran and N. Sriraam. Approximate entropy-based epileptic EEG detection using artificial neural networks. *IEEE Transactions on Information Technology in Biomedicine*, 11(3):288–295, 2007.

[37] V. Srinivasan, C. Eswaran, and N. Sriraam. Artificial neural network based epileptic detection using time-domain and frequency-domain features. *Journal of Medical Systems*, 29(6):647–660, 2005.

[38] S. Stober, A. Sternin, A.M. Owen, and A.G. Jessica Deep feature learning for EEG recordings. *arXiv preprint arXiv:1511.04306*, 2015.

[39] Y. Tang. Deep learning using linear support vector machines. *arXiv preprint arXiv: 1306. 0239*, 2013.

[40] L.T. Thanh. EEG epileptic spike detection using deep belief network, *Thesis*, 2016.

[41] E. D. Ubeyli and I. Güler. Features extracted by eigenvector methods for detecting variability of EEG signals. *Pattern Recognition Letters*, 28(5):592–603, 2007.

[42] W.R.S. Webber, B. Litt, K. Wilson, and R.P. Lesser. Practical detection of epileptiform discharges (EDs) in the EEG using an artificial neural network: A comparison of raw and parameterized EEG data. *Electroencephalography and Clinical Neurophysiology*, 91(3):194–204, 1994.

[43] D.F. Wulsin, J.R. Gupta, R. Mani, J.A. Blanco, and B. Litt. Modeling electroencephalography waveforms with semi-supervised deep belief nets: Fast classification and anomaly measurement. *Journal of Neural Engineering*, 8(3):036015, 2011.

[44] W. Zheng and B. Lu. Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks. *IEEE Transactions on Autonomous Mental Development*, 7(3):162–175, 2015.

[45] Epilepsy Foundation. URL http://www.epilepsy.com/learn

[46] WHO. URL http://www.who.int/mediacentre/factsheets/fs999/