Convolution Neural Network Hyperparameter Optimization for Vigilance Classification

Souhir Khessiba
Laboratoire de Technologie et Imagerie Médicale, Faculté de Médecine de Monastir, Université de Monastir, 5019
Monastir, Tunisie
ISITCom, Université de Sousse, 4011, Sousse, Tunisie
souhirkhessiba75@gmail.com

Ahmed Ghazi Blaiech
Laboratoire de Technologie et Imagerie Médicale, Faculté de Médecine de Monastir, Université de Monastir, 5019
Monastir, Tunisie
Institut Supérieur des Sciences Appliquées et de Technologie de Sousse, Université de Sousse, 4003
Sousse, Tunisie
ahmedghaziblaiech@yahoo.fr

Asma Ben Abdallah
Laboratoire de Technologie et Imagerie Médicale, Faculté de Médecine de Monastir, Université de Monastir, 5019
Monastir, Tunisie
Institut supérieur d'informatique et de Mathématiques, Université de Monastir, 5019
Monastir, Tunisie
assoumaba@yahoo.com

Khaled Ben Khalifa
Laboratoire de Technologie et Imagerie Médicale, Faculté de Médecine de Monastir, Université de Monastir, 5019
Monastir, Tunisie
Institut Supérieur des Sciences Appliquées et de Technologie de Sousse, Université de Sousse, 4003
Sousse, Tunisie
kbkhalifa@gmail.com

Antoine Manzanera
U2IS, ENSTA Paris, Institut Polytechnique de Paris,Palaiseau,France,91120
antoine.manzanera@ensta-paris.fr

Mohamed Hédi Bedoui
Laboratoire de Technologie et Imagerie Médicale, Faculté de Médecine de Monastir, Université de Monastir, 5019
Monastir, Tunisie
medhedi.bedoui@fmm.rnu.tn

***Abstract*—The performance of Convolution Neural Networks (CNN) is highly sensitive to the choice of the hyperparameters that define the structure of the network and the learning process. When facing a new application, tuning a deep neural network is a tedious and time-consuming process. As a result, the human expert through a slow trial and error process guided mainly by intuition does the choice of architecture manually. This explains the necessity of automating the calibration of these hyperparameters. In this paper, we outline and describe the use of hyper-parameter optimization (HPO) based algorithms in order to auto-design and train DL models for predicting individuals' vigilance states using an EEG signal. Particularly, we estimate the Tree Parzen Estimator (TPE) algorithm with Optuna framework on vigilance dataset. Experiments have shown that by utilizing the TPE optimization, we could achieve the best hyperparameter configuration, which allows us to learn and predict vigilance states more precisely. The accuracy and F1 score can reach 0.90 and 0.87, respectively, with the 1D-CNN-LSTM model using TPE, which demonstrates the effectiveness of the proposed method.**

Keywords— Hyperparameter optimization, Vigilance, Deep neural networks, EEG

I. Introduction

Electroencephalography (EEG) is the main modality for studying the electrical activity of the brain, and it has proven to be very suitable for predicting vigilance states. However, the classification of these states from this signal requires sophisticated approaches in order to achieve the best performance. Deep Learning (DL) approaches have shown a good performance in learning the high-level features of signals [1]-[2], particularly for EEG. Their large number of hidden layers that provide the most efficient solutions thanks to massive calculations characterizes them. One of the most powerful models in DL approaches is the Convolutional Neural Network (CNN). Thus, many studies have suggested CNN models for analyzing the EEG signal. In [3], the authors utilized the concept of DL on EEG signals to predict the driver's cognitive workload. A CNN model was used for extracting features and accurately classifying the cognitive workload. The conducted experimental results showed that the proposed system could provide an accurate classification of high and low cognitive workload sessions. In [5], the authors proposed two DL models to predict individuals' vigilance states based on the study of one derivation of EEG signals: a 1D-UNet model and 1D-UNet-Long Short-Term Memory (1D-UNet-LSTM). The experimental results showed that the suggested models could stabilize the training process and well recognize the subject vigilance. For example, the per-class average of precision and recall could be respectively up to 86% with 1D-UNet and 85% with 1D-UNet-LSTM. All these studies have used many DL approaches to analyze EEG signals, but the human expert through a slow trial and error process guided mainly by intuition has done the choice of the architecture empirically. In fact, the performance of these models is highly sensitive to the choice of the hyperparameters that define the structure of the network and the learning process. However, the rising popularity of DL models and their usage for diverse applications required the automatization of this process to adapt to each problematic. Much research has been done in the field of Hyperparameter Optimization (HPO), which can be categorized as being either mono-objective or multi-objective, such as grid search, random search, Bayesian optimization, and gradient-based optimization [6]–[8]. Grid search and manual search are the most widely used strategies for HPO [6]. These approaches make reproducibility harder and are impractical when there are a large number of hyperparameters. Thus, the idea of automating hyperparameter search has been increasingly researched. Thus, many authors have focused on further automating the calibration of hyper-parameters. In [9], a parallel version of the Particle Swarm Optimization (PSO) algorithm was proposed for the hyper-parameter optimization of DL models to overcome two problems: (i) the search space which is usually high dimensional, and (ii) the high runtime. The experiments have revealed that the PSO would largely take advantage of the rapidity offered by computational parallelization. Those DNN architectures that are optimized using parallel PSO offer a superior classification performance compared to those set up manually. The authors in [10] investigated lung nodule classification by proposing a multi-level CNN whose hyperparameter configuration was optimized by using a proposed Gaussian process with stationary kernels. The experiments demonstrated that the algorithm outperformed manual tuning. The authors in [11] put forward a new surrogate-based multi-objective optimization algorithm called the multi-objective Tree Parzen Estimator (TPE), which was an extension of the TPE widely used to solve expensive single-objective optimization problems. The experimental results showed that the suggested approach converged faster than the existing methods. The TPE success in expensive optimization problems indicates that it may outperform the existing methods [11]. Therefore, in this paper, we describe the use of the TPE algorithm for automatically designing and training DL models to predict individuals' vigilance states using an EEG signal. This algorithm is applied on the 1D-CNN-LSTM and 1D-CNN models to improve the classification performance. This paper is structured as follows: Section 2 presents the materials and methods and introduces the DL models successfully implemented for vigilance state classification. It also defines the TPE optimization algorithm. Section 3 presents the data and the experimentation setup. Moreover, this section describes the results of the optimized suggested model and elaborates the discussion based on the obtained results. The last section concludes the paper and gives some future perspectives.

II. MATERIALS AND METHODS

One of the most important strategies used to estimate vigilance consists in using physiological measures to give more precise data about the state of an individual. The sequential steps of the development of the automated vigilance state detection system are EEG data collection, pre-processing and classification by DL and using hyperparameter optimization (Figure 1).

A. EEG signal and preprocessing

The EEG signal is adopted in this paper to predict the vigilance states. This nonlinear and non-stationary signal characterizes the brain activity through a weakly invasive acquisition process, with electrodes placed along the scalp.

To prepare the dataset, we use the same subjects (S1, S2, and S3) as those collected in the experimentation of the previous work of our team [5] [12]. The EEG data are directly recorded from 28 active electrodes from the scalp at the Department of Functional Explorations of the Nervous System at Sahloul University Hospital, Tunisia. This signal is recorded during three 24-h periods with a 15-day interval, and it involves three healthy male subjects aged between 18 and 23. For each subject, the signal is recorded for two states: vigilance and drowsiness. The EEG recordings are done, reviewed and approved by an expert, in order to label the different levels of alertness. In this work, we focus on analyzing a single EEG signal from the right pariéto-occipital (Pz-Oz) electrode used to characterize analyzed vigilance states. This choice is justified by the fact that experts agree that this signal is the most appropriate to reflect the vigilance state [5] [12]-[14].

In the first step of preprocessing, we split the signal into periods of four seconds (recommended by an expert) in order to reduce the computation complexity. Then, we filter this signal to eliminate artifacts using a high-pass filter to filter out slow frequencies less than 0.1 Hz and a low-pass filter to filter out frequencies above 21Hz, for obtaining after that a good decision-making on the state of alertness.

The next step of preprocessing is the spectral analysis of the signal, which is described and successfully implemented in [5] [12]:

(i) The 512-point Fast Fourier Transform (FFT) is used to transform the acquired time-series EEG data into a frequency domain.

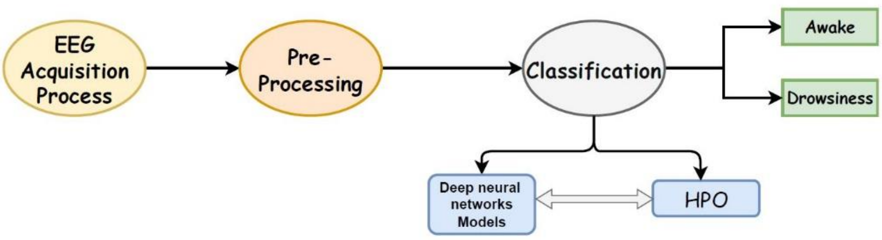

Fig.1. EEG signal processing steps with HPO of DL models for vigilance state classification

(ii) The frequency range [0.1, 21Hz], which is specific to the range of physiological waves is split into elementary frequency bands (one part for each frequency unit) to characterize this electrical activity.
(iii) In each band, the Spectral Band Power (PBS), which corresponds to the sum of the spectral amplitudes belonging to the spectral interval concerning the frequency band, is calculated.
(iv) The Percentage of the Relative Spectral Power (PPSR) of each band is computed, which is equal to the PBS divided by the total spectral power.

Computing the PPSRs relating to intervals u_i and u_{i+1} is given by equation (1):

$$PPSR_i = \frac{PBS_{[u_i, u_{i+1}]}}{TSP} \times 100; \qquad \begin{cases} u_i = 0.1 + (i-1) * \Delta u \; ; \; i \in [1...k] \\ \Delta u = \dfrac{(21-0.1)}{k} \end{cases} \tag{1}$$

Where Δu the length of the frequency band (Hz), k is the number of bands, and *TSP* represents the total spectral power. This means that the [0.1–21 Hz] interval is discretized into k regular sub-intervals of length Δu. Thereby, the PPSR will be the input to the classification tool for vigilance state detection for each four seconds.

B. DL models and hyperparametrers

DL models are widely applied to various areas like computer vision, classification and segmentation, since they have had great success solving many types of highly complex problems.

Among the most powerful models in DL approaches are the CNN, in particular the 1D-CNN, which has been well adopted in the literature for processing EEG signals [15, 16]. Its architecture is usually composed by a series of 1D convolutional, pooling, normalization and fully connected layers. In this paper, we use the DL models [5] that were implemented for vigilance state classification: 1D-CNN and 1D-CNN-LSTM.

1) *1D-CNN:* it uses a system that has been designed for reduced processing requirements through convolution layers that utilize a single weight assigned to the inputs in the same convolution filter for all neurons. The convolution layers can adaptively learn informative features in the input signal. The low-level representations of the input data are learned by the early layers and then passed into later layers to hierarchically learn the high-level representations. The CNN may be used to classify several dimensionalities of inputs (1D, 2D,..). The 1D-CNN has been well adopted in the literature for the treatment of EEG signals. Its architecture is commonly composed by a series of 1D convolutional, pooling and fully connected layers. This architecture can include also normalization layers.

2) *1D-CNN-LSTM:* This model, presented in Figure 3, is a combination between 1D-CNN and LSTM. The LSTM model is well known for being capable of learning the problem of long-term dependencies in temporal data, given that it includes an input that depends on the previous computation. The 1D-CNN-LSTM architecture takes the output of 1D-CNN (last layer) to feed in as the input of the LSTM network. This latter is made up of four hidden cells, where a dropout layer to prevent overfitting follows each cell. At the end, the 1D-CNN-LSTM architecture integrates a batch normalization layer, a fully connected layer and Softmax layers to accomplish the classification task.

3) *Hyperparameters:* The DL models have many hyperparameters, including those that specify the structure of the network itself and those that determine how the network is trained. As the training speed of these networks is slow, it is difficult to adjust the hyperparameters. When training a network, the result of the model will depend not only on the chosen structure but also on the training method, which itself has several hyperparameters such as the learning rate, the loss function, the mini-batch size, and the number of training iterations. Furthermore, the structure of the neural network itself involves numerous hyperparameters in its design, including the size of each layer, the number of hidden layers, the number of convolution layers, the kernel size, the filter size, the activation function, the weight initialization, etc.

Table 1 summarizes the hyperparameters responsible for defining the structure of the network and those related to the optimization and training process. Tuning the hyperparameters of DL models is a critical and time-consuming process that has been mainly done relying on the knowledge of the experts. This explains the necessity of automating the calibration of these hyperparameters.

C. HPO algorithm

DL algorithms have been used widely in various applications and areas. To fit a deep learning model into different problems, its hyperparameters must be tuned. Selecting the best hyperparameter configuration for deep learning models has a direct impact on the model performance.

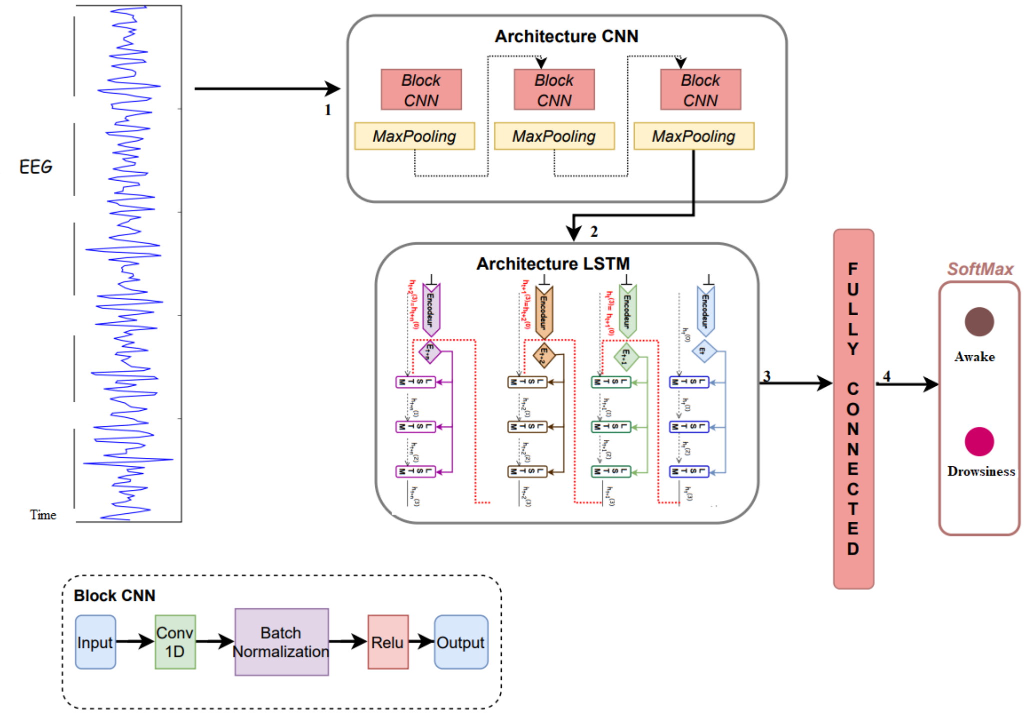

Fig.2. 1D-CNN-LSTM architecture

TABLE 1: HYPERPARAMETERS DEFINING ARCHITECTURE AND TRAINING PROCESS OF NEURAL NETWORK.

Hyperparameters	**Type**	**Scope**
Number of convolution layers	Categorical	{0,1,..,25}
Number of LSTM layers	Categorical	{0,1,..,25}
Number of dense layers	Categorical	{0,1,..,25}
LSTM units	Categorical	{32,,..,512}
Optimizer	Categorical/Integer	{Adam, Rmsprop, Adadelta }
Filter size	Integer	{64,128,..,1024}
Kernel Size	Integer	{0,…..,10}
Batch size	Integer	{10, 32, 64,128}
Learning rate	Float	{0;1}
Dropout rate	Float	{0;1}
Activation function	Categorical/Integer	{Relu, Sigmoid, Tanh}

It often requires deep knowledge of algorithms and appropriate hyperparameter optimization techniques. Several methods have been proposed for HPO such as grid search [6], random search [8], simulated annealing [17] and Bayesian optimization [18], and TPE [11][19]. The TPE success in expensive optimization problems indicates that it may outperform existing methods [11]. This algorithm is a Sequential Model-Based Optimization (SMBO) approach. SMBO methods sequentially construct models to approximate the performance of hyperparameters based on historical measurements, and then subsequently choose new hyperparameters to be tested based on this model.

Consequently, the TPE is an iterative process that uses the history of evaluated hyperparameters to create a probabilistic model, which is used to suggest the next set of hyperparameters to evaluate.

Let assume a set of observations that takes $\{(x^{(1)}, y^{(1)}),, (x^{(k)}, y^{(k)})\}$. To apply the TPE, the observation results are divided into good and poor results by a pre-defined percentile y^*. The TPE defines p(x, y) using the following two probability density functions given by equation (2):

$$P(x|y) = \begin{cases} l(x) & if\ y < y^* \\ g(x) & if\ \ y \geq y^* \end{cases} \qquad (2)$$

Where $l(x)$ is the probability density function formed using the observed variables $\{x^{(i)}\}$ such that $y^* > y^{(i)}$ $(= f(x^{(i)}))$, and $g(x)$ is the probability density function using the remaining observations. Value y^* is selected to be a quantile γ of the observed y values satisfying $p(y^* > y) = \gamma$
After that, the expected improvement in the acquisition function is reflected by the ratio between the two density functions, which is used to determine the new configurations for evaluation.

III. EXPERIMENTS AND RESULTS

The implementation has been done to show the effectiveness of the HPO algorithm used to improve the performance of vigilance state classification.

A. *Experiment setting*

We evaluate the hyperparameter optimization algorithm on the DL architectures including 1D-CNN and 1D-CNN-LSTM. Those architectures are developed using Keras whose libraries are written in Python. The experiments are achieved with an experimental implementation on a Portable Gaming PC with an Intel 9th-generation Core i5-9300H processor, a NVIDIA GeForce GTX 1650 Graphics card and 8 GB Memory.

To tackle HPO problems, we use Optuna framework [20], which provides many HPO algorithms including the TPE.

B. *Results and discussion*

This section describes the results obtained through our experimentation using the HPO TPE approach. We focus on three subjects [5] [12] with the same size of observations in order to detect the vigilance states.

The within-subject vigilance state classification is applied to evaluate the performance by different models, where each subject is taken separately and divided into 80% and 20% of observations for training and testing, respectively.

Table 2 presents the hyper-parameter values obtained by the implemented DL models for the three subjects. This table shows that Adam function is more selected as an optimizer, which justifies the effectiveness of this function. Furthermore, the ReLU function is selected for all implementations. We note that the hyper-parameter values change between the models for the same subject. This proves that the hyperparameters are specific to the utilized architectures. Furthermore, the hyperparameter values vary between the subjects with the same DL model. This proves also that the hyperparameters depend on the input data, even if working in the same context.

Table 3 exposes the accuracy results obtained using HPO and compared with the results before the optimization process for the 1D-CNN, 1D-CNN-LSTM models. We note that the classification performance in terms of accuracy is good with HPO. Accuracy for subject S1 using HPO can be up to 0.89 and 0.90 with 1D-CNN and 1D-CNN-LSTM, respectively.

Table 4 describes the classification performance in terms of recall, precision and F1-score using DL architectures for subject S1, which has the best classification accuracy, as depicted in Table 3 (the per-model average is 0.89). This table shows that the precision can achieve 0.88 using 1D-CNN-LSTM with HPO. The F1-score can be up to 0.85 using 1D-CNN with HPO compared to the same model without optimization.

Given Table 3 and Table 4, we note that including an optimization phase of hyperparameters allows to significantly improving the classification performance for all subject and for all implemented DL models. Indeed, these results show that the iterative process of TPE is suitable for our application.

IV. CONCLUSION AND PERSPECTIVES

In this paper, we have introduced and explored the potential of HPO algorithms in order to give the best configurations of hyperparameters and to improve the performance of vigilance state classification based on the analysis of cerebral activities using EEG signals. The HPO TPE has been applied to the 1D-CNN and 1D-CNN-LSTM models, and the optimal hyper-parameter configuration has been generated. The experimental results in the study have revealed that the performance of vigilance state classification has been improved using the HPO TPE method.

In the future, we will add more subjects for further validation of the DL architectures with hyperparameter optimization. In addition, we will evaluate more HPO algorithms in order to improve the system performance.

TABLE 2 : BEST HYPERPARAMETERS CONFIGURATIONS USING TPE ALGORITHM

	1D-CNN			1D-CNN-LSTM		
	S1	**S2**	**S3**	**S1**	**S2**	**S3**
Number of convolution layers	9	10	20	10	13	10
Number of LSTM layers	-	-	-	3	5	4
Number of dense layers	-	-	-	-	-	-
LSTM units	-	-	-	100	64	512
Optimizer	Rmsprop	Adam	Adam	Adam	Adam	Adam
Filter size	32	64	64	32	32	64
Kernel size	1	1	1	1	1	1
Batch size	64	32	64	10	10	10
Learning rate	0.001	0.001	1e-05	0.001	0.003	0.001
Dropout rate	0.2	0.4	0.1	0.3	0.5	0.1
Activation function	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU

TABLE 3 : SUBJECT VIGILANCE STATE CLASSIFICATION ACCURACY

	S1		S2		S3	
	Without HPO[5]	**HPO**	**Without HPO [5]**	**HPO**	**Without HPO [5]**	**HPO**
1D-CNN	0.80	0.89	0,79	0.88	0.80	0.82
1D-CNN-LSTM	0.84	0.90	0,73	0.85	0.76	0.85
AVG/Model	**0.82**	**0.89**	**0.76**	**0.86**	**0.78**	**0.83**

TABLE 4 : PERFORMANCE MEASURES OF PROPOSED MODELS FOR SUBJECT 1

	Recall		Precision		F1-Score	
	Without HPO [5]	**HPO**	**Without HPO [5]**	**HPO**	**Without HPO [5]**	**HPO**
1D-CNN	0.77	0.84	0.86	0.87	0.77	0.85
1D-CNN-LSTM	0.85	0.87	0.80	0.88	0.81	0.87

REFERENCES

[1] Boudegga, H, .Elloumi, Y.,Akil, M., Bedoui, M.H., Kachouri, K.,Ben Abdallah, A.,« Fast And Efficient Retinal Blood Vessel Segmentation Method Based On Deep Learning Network », Computerized Medical Imaging And Graphics, Mars (2021)

[2] Y.Lecun, Y.Bengio, G. Hinton,"Deep learning". Nature. Volume 521:page436–444, 2015.

[3] M.A.Almogbel, A.H .Dang, W.Kameyama, "EEG-Signals Based Cognitive Workload Detection of Vehicle Driver using Deep Learning". International Conference on Advanced Communications Technology(ICACT), Page: 256-259, 2018.

[4] M.Hajinoroozi, J.Zhang, Y. Huang, "Driver's Fatigue Prediction by Deep Covariance Learning from EEG". IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2019.

[5] S.Khessiba, A.G. Blaiech, K.Ben Khalifa, A.Ben Abdallah, M.H.Bedoui "Innovative deep learning models for EEG-based vigilance detection", Neural Computing and Applications, 33, pages6921–6937, 2021.

[6] M. Claesen et B. De Moor, « Hyperparameter Search in Machine Learning », arXiv:1502.02127 [cs, stat], avr. 2015, Consulté le: juill. 03, 2021.

[7] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, et F. Hutter, « Auto-sklearn: Efficient and Robust Automated Machine Learning », in Automated Machine Learning, F. Hutter, L. Kotthoff, et J. Vanschoren, Éd. Cham: Springer International Publishing, p. 113-134, 2019.

[8] J. Bergstra et Y. Bengio, « Random Search for Hyper-Parameter Optimization », Journal of Machine Learning Research 13 281-305 p. 25, 2012.

[9] R. Brito, S. Fong, Y. Zhuang, et Y. Wu, « Generating Neural Networks with Optimal Features through Particle Swarm Optimization », in Proceedings of the International Conference on Big Data and Internet of Thing - BDIOT2017, London, United Kingdom, p. 96-101. 2017,

[10] W.H.Khoong, A Heuristic for Efficient Reduction in Hidden Layer", n book: Intelligent Computing, pp.208-218, 2020.

[11] Y. Ozaki, Y. Tanigaki, S. Watanabe, et M. Onishi, « Multiobjective tree-structured parzen estimator for computationally expensive optimization problems », in Proceedings of the 2020 Genetic and Evolutionary Computation Conference, Cancún Mexico, p. 533-541, juin 2020,

[12] A.G Blaiech, K.Ben Khalifa, M. Boubaker, M.HB edoui, "LVQ neural network optimized implementation on FPGA devices with multiple-wordlength operations for real-time systems. Neural Computing and Applications, volume 29, pages509–528, 2018.

[13] K.Ben Khalifa, MH .Bedoui, M .Dogui, F .Alexandre, "Analysis of vigilance states by neural networks". In: Proceedings of international conference on information and communication technologies: from theory to applications, pp 429–430, 2004.

[14] AG. Blaiech, K.Ben Khalifa, M Boubaker, MH.Bedoui, "Multi-width fixed-point coding based on reprogrammable hardware implementation of a multi-layer perceptron neural network for alertness classification". In: Proceeding of the 10th international conference on intelligent systems design and applications (ISDA), Cairo, Egypt, 2010, pp 610–614., 2010.

[15] S.Kiranyaz, O.Avci, O.Abdeljaber, T.Ince, M.Gabbouj, D.J.Inman , "1D Convolutional Neural Networks and Applications: A Survey, In arXiv:1905.03554. 2019.

[16] O.Yıldırım, U.BB aloglu, U.R Acharya, A deep convolutional neural network model for automated identification of abnormal EEG signals, Neural Computing and Applications. Published: 23 November 2018. 2018.

[17] V. Černý, « Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm », J Optim Theory Appl, vol. 45, no 1, p. 41-51, janv. 1985, doi: 10.1007/BF00940812.

[18] J.noek , H.Larochelle, R. P. Adams « Practical Bayesian optimization of machine learning algorithms", | Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2 », 2012

[19] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kegl, Algorithms for hyperparameter optimization, Proc. Adv. Neural Inf. Process. Syst.2011

[20] T.Akiba , S.Sano , T.Yanase , T. Ohta , and M.Koyama, « Optuna: A Next-generation Hyperparameter Optimization Framework », in https://arxiv.org/pdf/1907.10902.pdf, 2020.