CentraleSupélec

Machine Learning et Classification

Challenge de classification de phases du sommeil

Option Mathématiques Appliquées

Année scolaire 2019–2020

Henrique Miyamoto
henriquekoji.miyamoto@supelec.fr

6 janvier 2020

Table des matières

1 Introduction

Le sommeil joue un rôle fondamental pour la santé d'un individu. Il peut être divisé dans cinq phases (*stages*), à savoir : éveil, sommeil léger 1 (NREM1), sommeil léger 2 (NREM2), sommeil profond (NREM3) et sommeil paradoxal (REM). Chaque phase est associée à une fonction différente, d'où l'importance de surveiller ces phases pour diagnostiquer des troubles du sommeil [1]. Traditionnellement, la classification des phases du sommeil est faite manuellement par un spécialiste, à partir des données d'un examen de polysomnographie. Cependant ce travail est répétitif et soumis à des divergences entre spécialistes, ce qui motive l'étude des méthodes automatiques de classification [2].

Le but de ce *challenge* est alors de proposer et implémenter une méthode automatique pour la classification des phases du sommeil à partir des données de trois types de capteurs : sept électroencéphalogrammes (EEG), un oxymètre de pouls et un accéléromètre de trois axis. Les signaux sont divisés en fenêtres de 30 s et ont été échantillonnés à 50 Hz (EEG) ou 10 Hz (oxymètre de pouls et accéléromètre). Le jeux de données de entraînement contient 24688 donnés classifiés et le jeux de données de test contient 24980 donnés à classifier.

On propose dans ce travail une approche simple, basée sur [2, 3, 4], où on extrait des caractéristiques temporelles et fréquentielles de chaque signal, lesquelles on utilise comme variables explicatives dans un modèle d'apprentissage statistique de type *random forest*. L'implémentation a été faite avec R (version 3.6.2). Les codes R sont disponibles sur `https://github.com/miyamotohk/sleep_stage_classification`.

2 Visualisation des données

D'abord, on présente une visualisation des données. Pour cela, on trace la moyenne et les bornes supérieur et inférieur de chaque signal pour chaque phase (Figures 2.1 et 2.2).

3 Méthodes

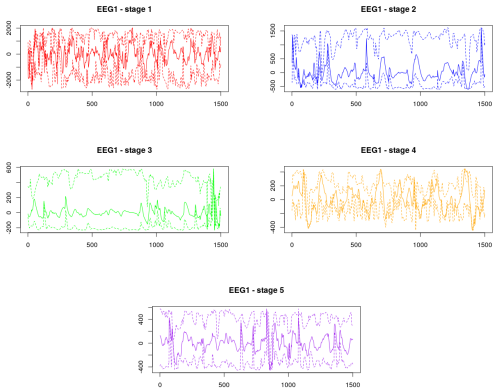
La principal *feature* (caractéristique) utilisée dans la littérature pour distinguer les phases du sommeil sont les signaux EEG (cf. [2, Table 13]). Chaque signal EEG peut être décomposée en bandes de fréquence, qui portent des informations utiles pour la classification des phases du sommeil. La définition des fréquences pour chaque bande peut varier légèrement dans la littérature (voir par exemple [2] contre [3]). Dans ce travail, on utilise les définitions comme dans la Table 3.1.

Table 3.1 – Bandes de fréquence d'un signal EEG décomposé.

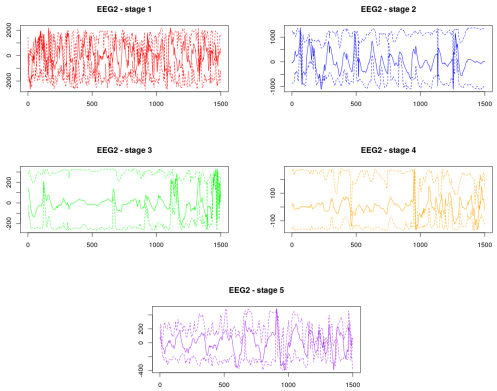
Bande	**Fréquences (Hz)**
Delta (δ)	0.5–4
Theta (θ)	4–8
Alpha (α)	8–13
Beta (β)	13–25
K-complexe (K)	0.9–1.1

En plus des données EEG, on dispose aussi des signaux de l'oxymètre et de l'accéléromètre, qui sont plus rares dans la littérature.

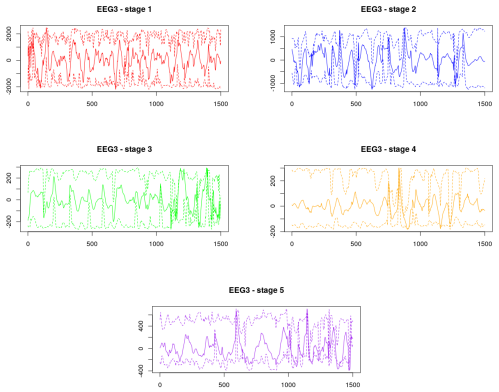
D'abord, on divise les signaux en deux groupes, à savoir : signaux EEG et signaux oxymètre et



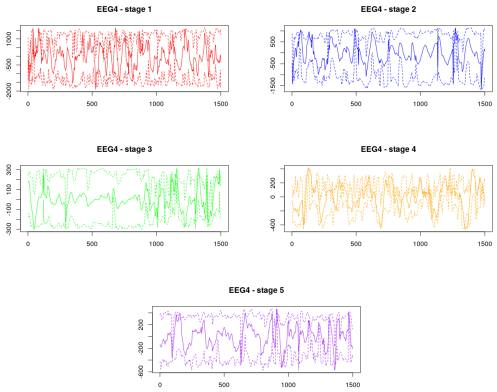
(a) Signal EEG1 pour chaque phase.



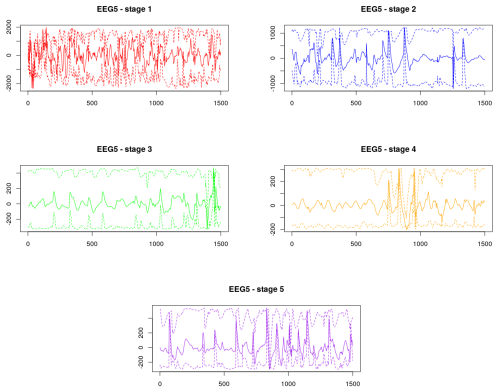
(b) Signal EEG2 pour chaque phase.



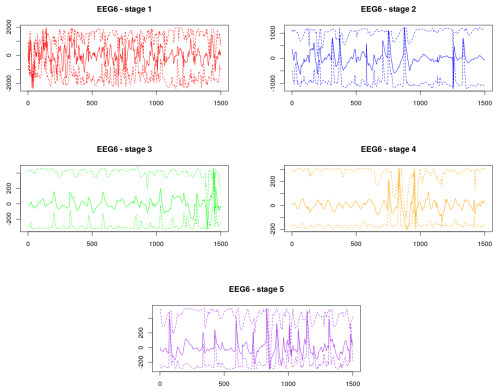
(c) Signal EEG3 pour chaque phase.



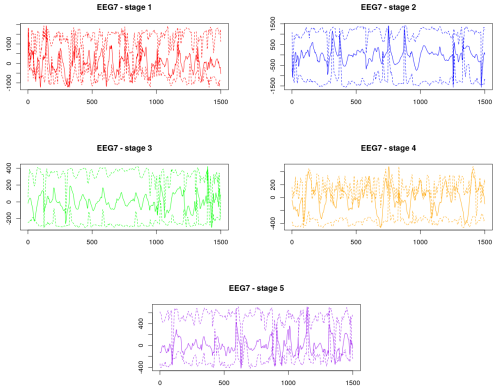
(d) Signal EEG4 pour chaque phase.



(e) Signal EEG5 pour chaque phase.

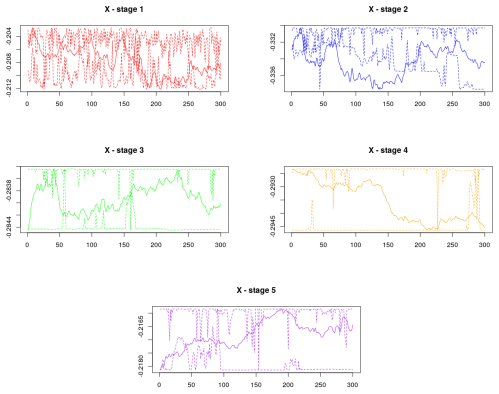


(f) Signal EEG6 pour chaque phase.

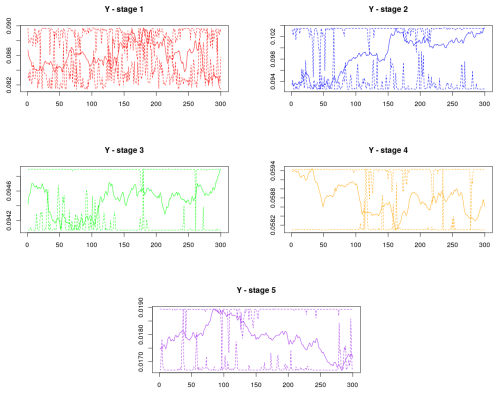


(g) Signal EEG7 pour chaque phase.

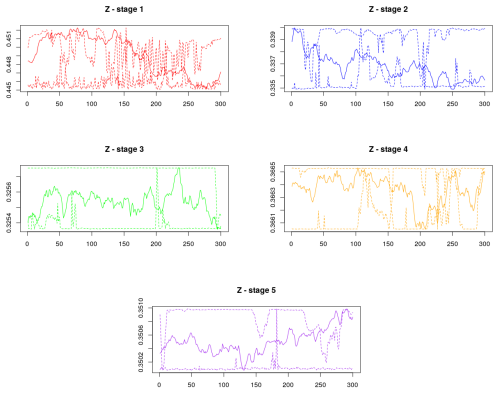
FIGURE 2.1 – Visualisation des signaux EEG.



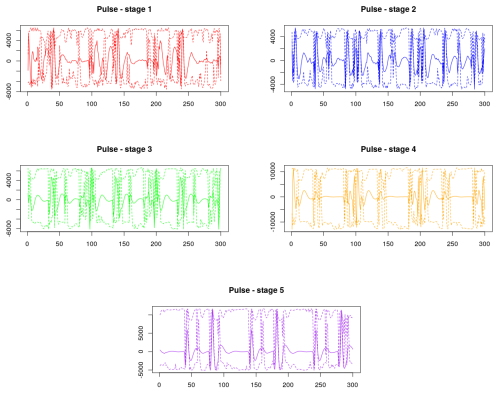
(a) Signal axis X pour chaque phase.



(b) Signal axis Y pour chaque phase.



(c) Signal axis Z pour chaque phase.



(d) Signal oxymètre pour chaque phase.

FIGURE 2.2 – Visualisation des signaux de l'accéléromètre et de l'oxymètre de pouls.

accéléromètre. Pour chaque groupe, on propose des *features* de deux types : du domaine du temps et du domaine de la fréquence, comme dans la Table 3.2.

TABLE 3.2 – Résumé des *features* extraites.

Groupe	***Feature***	**Symbole**
Signaux EEG	*Domaine du temps*	
	Moyenne	μ
	Variance	σ^2
	Zero crossing rate moyen	Z_c
	Domaine de la fréquence	
	Puissance spectrale relative	$\delta_r, \theta_r, \alpha_r, \beta_r, K_r$
	Ratios de puissance	$\delta/\alpha, \theta/\alpha$
	Pic de fréquence	sp
Signaux oxymètre/accéléromètre	*Domaine du temps*	
	Moyenne	μ
	Variance	σ^2
	Skewness	γ
	Kurtosis	κ
	Zero crossing rate moyen	Z_c
	Domaine de la fréquence	
	Puissance spectrale relative	L_r, H_r
	Puissance spectrale totale	P
	Pic de fréquence	sp

3.1 Prétraitement des données

Pour pouvoir extraire les *features* du domaine de la fréquence, il est nécessaire d'estimer le diagramme de densité spectrale de puissance (DSP), ce qui est fait avec la méthode d'estimation de Welch [5]. Cette méthode consiste à calculer la moyenne entre transformées de Fourier consécutives de petites fenêtres du signal et améliore la précision du périodogramme classique dans le cas des signaux non-stationnaires [6]. L'implémentation en R est fait avec la commande `pwelch`. Notons que, par défaut, le périodogramme est calculé entre zéro et la moitié de la fréquence d'échantillonnage du signal.

Ensuite, pour calculer les puissances spectrales, on intègre la courbe de densité spectrale de puissance obtenue entre les fréquences souhaitées, ce qui peut être fait avec la méthode `integrate.xy`.

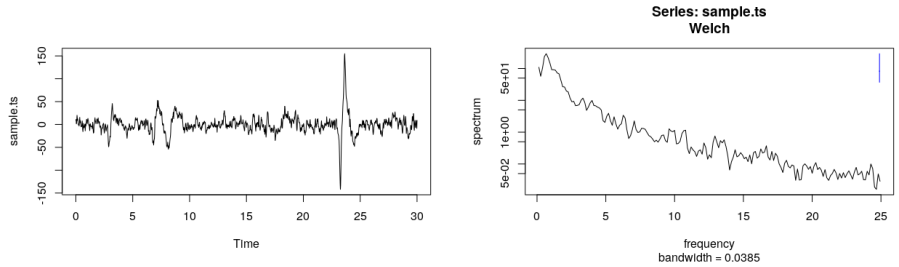


FIGURE 3.1 – Exemple de signal (EEG1, échantillon 1000) et son estimation de Welch de la DSP.

3.2 Extraction de *features*

La choix des *features* était basée sur [3, 4]. Pour les signaux EEG, les *features* principales sont dérivées de la puissance spectral des bandes $\delta, \theta, \alpha, \beta, K$ [3]. Pour les signaux oxymètre et accéléromètre, on propose des *features* statistiques classiques [4] et la puissance spectrale dans la zone haute et basse du spectre.

3.2.1 Domaine du temps

Pour extraire les *features* du domaine du temps, considérons les signaux comme une série chronologique $\{x_1, x_2, \ldots, x_N\}$.

— La **moyenne** μ est calculée comme

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

— La **variance** σ^2 du signal est calculée comme

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu)^2.$$

— Le **skewness** γ est une mesure de l'asymétrie de la distribution autour de la moyenne [4] :

$$\gamma = \frac{1}{N}\sum_{i=1}^{N} \left(\frac{x_i - \mu}{\sigma}\right)^3.$$

— La **kurtosis** κ est un coefficient d'acuité de la distribution [4] :

$$\kappa = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \mu}{\sigma}\right)^4.$$

— Le **zero crossing rate moyen** Z_c est la moyenne du taux de changement de signe d'un signal dans fenêtres de taille $\frac{3}{2}F_s$, où F_s est la fréquence d'échantillonnage [3].

3.2.2 Domaine de la fréquence

— Pour les signaux EEG, les **puissances spectrales relatives** $\delta_r, \theta_r, \alpha_r, \beta_r, K_r$ on été obtenues en divisant la puissance de chaque bande (voir Table 3.1) par la puissance totale. Pour les signaux oxymètre et accéléromètre, on calcule la puissance relative dans zones du spectre : basse L_r (0–2.5 Hz) et haute H_r (2.5–5 Hz).

— La **puissance spectrale totale** P est obtenue en intégrant la DSP dans tout le domaine.

— Les **ratios de puissance** δ/α et θ/α sont obtenus en divisant les puissances relatives des bandes concernées [3].

— Le **pic de fréquence** *sp* est la fréquence pour laquelle la DSP est maximale [3].

3.3 Modèle

On utilise le modèle *random forest* [7, Chapitre 15] pour l'apprentissage statistique. Il s'agit d'une modification de la méthode *bagging* qui construit une grande collection d'arbres décorrélées, présenté originalement dans [8] et qui est devenu assez populaire, grâce à ses avantages – notamment une bonne performance avec très peu de réglages.

L'idée fondamentale du *bagging* c'est de prendre la moyenne de plusieurs modèles bruités, mais approximativement non-biaisés, de sorte à réduire la variance. Ainsi, les arbres sont des candidats appropriés pour l'appliquer, car ils peuvent capturer des interactions complexes dans les données et, si assez profondes, son biais est faible. En plus, comme les modèles générés sont bruités, ils peuvent profiter de la prise de moyenne.

Pour encore réduire la variance du modèle de *bagging*, la méthode *random forest* propose une choix aléatoire des variables d'entrée lors de la construction des arbres. Plus spécifiquement, quand on construit un arbre sur un jeux de données, avant chaque division, on choisit $m \leq p$ données d'entrée au hasard comme candidats pour la division. Après B tels arbres $\{T_b(x)\}_{b=1}^{B}$ sont construits :

— Pour faire de la *régression*, la prédiction du *random forest* est calculée comme

$$\hat{f}_B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x).$$

— Pour faire la *classification*, soit $\hat{C}_b(x)$ la classe prédite par le b-ème arbre. Alors, $\hat{C}_B(x)$ est le vœux majoritaire dans les $\{\hat{C}_b\}_{b=1}^{B}$.

Dans le cas de la classification, la valeur par défaut de m est $\lfloor\sqrt{p}\rfloor$ et la taille minimale d'un nœud est $n_{\min} = 1$. L'Algorithme 1 décrit de manière résumée la méthode *random forest* pour la classification.

L'implémentation du *random forest* sur R est faite avec le *package* `randomForest` [9]. On a utilisé les paramètres suivantes :

— Nombre d'arbres : 1000.

Algorithm 1: *Random forest* pour la classification [7, p. 588]

for $b = 1 : B$ **do**
  Prendre un échantillon *bootstrap* de taille N à partir des données de entraînement.
  Construire un arbre T_b en répétant récursivement, jusqu'à que la taille de nœud minimale est atteinte :
    1. Séléctionner m des p variables au hasard.
    2. Séléctionner la meilleure variable/point de division entre les m.
    3. Diviser le nœud en deux fils.
end
Retourner l'ensemble de arbres $\{T_b\}_{b=1}^{B}$.
La classe prédite $\hat{C}_B$ est le vœux majoritaire entre les $\{\hat{C}_b\}_{b=1}^{B}$, où $\hat{C}_b$ est la classe prédite par T_b.

— Nombre de variables sectionnées à chaque division : $m = 15$

— Taille minimale d'un nœud : $n_{\min} = 1$ (défaut)

3.4 Validation croisée

Pour estimer l'erreur de prédiction de la méthode *random forest*, on peut utiliser l'estimation d'erreur *out-of-bag* (OOB) : pour chaque observation $z_i = (x_i, y_i)$, on construit la prédiction *random forest* en utilisant seulement les arbres qui correspondent à des échantillons *bootstrap* dans lesquels z_i *n'est pas* utilisé. Ensuite, on calcule la moyenne de l'erreur de prédiction de ces modèles pour obtenir l'estimation d'erreur OOB. Notons que cette estimation est "presque identique" à l'estimation obtenue par validation croisée de type *N-fold* [7].

4 Résultats et discussion

4.1 Erreur de classification

La matrice de confusion du modèle générée est présenté dans Table 4.1. On observe que la classification des phases du sommeil est raisonnable pour tous les phases, à l'exception de la phase 1 (NREM1), qui est la la plupart du temps mal classifiée comme phase 2 (NREM2), phase 0 (éveil) ou phase 4 (REM).

Table 4.1 – Matrice de confusion.

	0	1	2	3	4	Erreur de classification
0	**2885**	42	380	57	147	0,1782968
1	402	**193**	719	20	337	0,8845003
2	234	32	**8303**	380	500	0,1212827
3	110	1	728	**4354**	31	0,1665391
4	231	50	823	79	**3650**	0,2447755

L'évolution de l'erreur de prédiction OOB est présenté dans Figure 4.1. La ligne noire représente l'erreur générale et chaque couleur la classification d'une phase. Comme on avait déjà noté avec la matrice de confusion, il y a seulement une phase qui est souvent mal classifiée (en vert dans le graphe, cela doit correspondre à la phase 1/NREM1). La valeur finale de l'erreur est de 21,48%. On observe aussi qu'il ne vaut pas la peine de trop augmenter le nombre d'arbres, car l'erreur de classification se stabilise à partir d'approximativement 100 arbres.

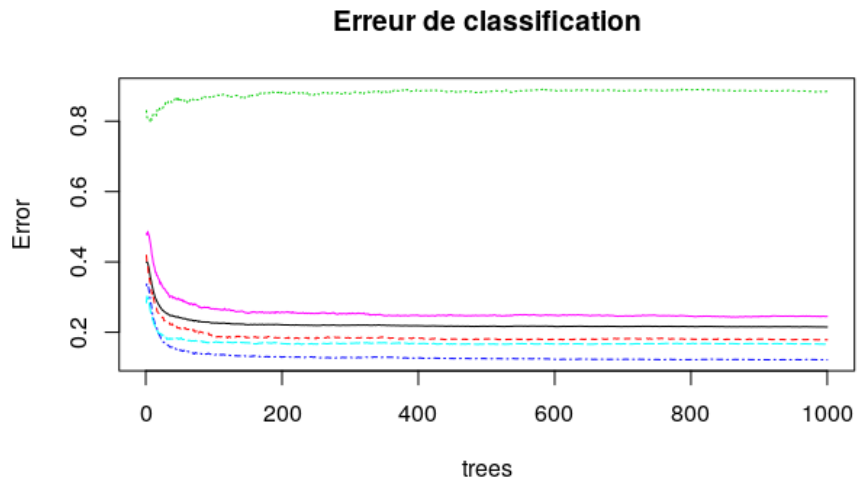


FIGURE 4.1 – Estimation de l'erreur OOB en fonction du nombre de arbres (ligne noire : général, lignes en couleur : pour chaque phase).

Enfin, on note que l'erreur de 21,48% obtenu avec *random forest* est plus petit que l'erreur de 25,03% obtenu avec la méthode SVM (noyau gaussien) pour les mêmes données et que la figure de mérite du challenge est le *mean F1-Score*, qui dépend de la précision p et du rappel r :

$$F_1 = 2\frac{pr}{p+r},$$

où $p = t_p/(t_p + f_p)$ et $r = t_p/(t_p + f_n)$, avec t_p nombre de vraies positives, f_p faux positives et f_n faux négatives. L'estimation du *F1-score* calculée par Kaggle était d'approximativement 0,68388 (calculée avec environ 70% des données).

4.2 Importance des variables

Le package `randomForest` permet extraire aussi l'importance des variables explicatives, calculées à partir de la diminution moyenne de l'impureté des nœuds. La Table 4.2 liste les 10 *features* le plus importantes selon cette classification.

TABLE 4.2 – Les 10 *features* le plus importantes.

#	*Feature*
1	theta/alpha_eeg_4
2	var_eeg_5
3	var_eeg_1
4	var_y
5	var_x
6	delta_eeg_5
7	zcmean_eeg_5
8	var_eeg_6
9	kur_z
10	theta/alpha_eeg_5

On note 7/10 variables sont des *features* liées aux signaux EEG et 3/10 sont des signaux de l'accéléromètre. On souligne l'apparition de variables statistiques (variance et kurtosis) et de variables liées à la puissance spectrale ($\theta/\alpha, \delta_r$).

Références

[1] Kaggle. *Dreem 2 Sleep Stage Classification Challenge*. Disponible sur : `https://www.kaggle.com/c/dreem-sleep-stages-2020/`.

[2] K. Aboalayon, M. Faezipour, W. Almuhammadi et S. Moslehpour. "Sleep Stage Classification Using EEG Signal Analysis : A Comprehensive Survey and New Investigation". *Entropy*, vol. 18, no. 9, p. 272, Août 2016.

[3] M. Radha, G. Garcia-Molina, M. Poel et G. Tononi. "Comparison of Feature and Classifier Algorithms for Online Automatic Sleep Staging Based on a Single EEG Signal". *Proceedings of 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Chicago, États-Unis, pp. 1876–1880, Août 2014.

[4] A.R. Hassan, S.K. Bashar et M.I.H. Bhuiyan. "On the Classification of Sleep States by Means of Statistical and Spectral Features from Single Channel Electroencephalogram". *IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Kochi, Inde, pp. 2238–2243, Août 2015.

[5] P. Welch. "The Use of Fast Fourier Transform for the Estimation of Power Spectra : A Method Based on Time Aver. aging Over Short, Modified Periodograms". *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, Juin 1967.

[6] R. Vallat. *Compute the average bandpower of an EEG signal*. Disponible sur : `https://raphaelvallat.com/bandpower.html`.

[7] T, Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning* : Data Mining, Inference and Prediction, 2nd edition. New York : Springer, 2017.

[8] L. Breiman. "Random Forests". *Machine Learning*, vol. 45, no. 1, pp. 5–32, Octobre 2001.

[9] R Documentation. *randomForest v4.6-14*. Disponible sur : `https://www.rdocumentation.org/packages/randomForest/versions/4.6-14`.