

Child Mind Institute - Détecter les états de sommeil
Détectez l'endormissement et le réveil à partir des données
d'un accéléromètre porté au poignet

Rapport projets Kaggle

2023 / 2024

Meryam BOULAYAT, Shahina MOHAMED, Mélanie GOU, Lounes
Mechouek et Chlomite COHEN

-

Master 2 Biologie-Informatique

-

Sommaire

Introduction.....	3
Matériel et méthodes.....	3
Jeu de données.....	3
Méthodes d'apprentissage automatique traditionnelles	4
Random Forest.....	3
Architectures de réseaux neuronaux.....	5
LSTM.....	5
CNN LSTM	4
GRU	6
Résultats	6
Traitement de données ou " <i>data processing</i> "	6
Méthodes d'apprentissage automatique traditionnelles	8
Random Forest.....	8
Architectures de réseaux neuronaux.....	9
LSTM.....	9
CNN LSTM	6
GRU	6
Conclusion	14
Bibliographie.....	15

Introduction

Le sommeil est un élément essentiel de la vie humaine, influençant divers aspects de notre bien-être, de nos fonctions cognitives à notre stabilité émotionnelle. Cependant, la compréhension des schémas complexes du sommeil demeure un défi, en grande partie en raison du manque de données naturelles associées à des annotations précises.

L'objectif de notre projet, qui s'inscrit dans le cadre d'une compétition proposée par la plateforme *Kaggle*, est de relever ce défi. Le but étant de sensibiliser à l'importance du sommeil, tout en explorant son lien avec l'environnement et le bien-être des enfants et des adolescents. Le succès de cette compétition revêt une grande importance pour ces derniers, en particulier ceux qui luttent contre des problèmes d'humeur et de comportement. En effet, le sommeil joue un rôle central dans la régulation de l'humeur, des émotions et du comportement, surtout au cours des années de développement. L'identification précise des périodes de sommeil et d'éveil offrirait donc aux chercheurs un outil puissant pour explorer plus en profondeur les schémas de sommeil et les perturbations chez les enfants. Et cela, a le potentiel de révolutionner la manière dont les interventions spécifiques sont conçues et mises en œuvre pour ces jeunes, notamment en partenariat avec l'Institut Child Mind (CMI) et le Centre mondial SNF pour la santé mentale des enfants et des adolescents.

À partir d'ensemble de données comprenant environ 500 enregistrements sur plusieurs jours de données d'accéléromètre porté au poignet, notre projet consistera à déterminer l'état de sommeil d'une personne. Dans ce but nous avons développé plusieurs modèles de prédiction répartis en deux classes. D'une part, nous avons utilisé une méthode d'apprentissage automatique traditionnelle appelé *Random Forest*. Puis, nous avons établis des architectures de réseaux neuronaux appelées *LSTM*, *CNN* avec *LSTM* et *GRU*.

Matériels et méthodes

Présentation du jeu de données

L'ensemble des données comprend environ 500 enregistrements sur plusieurs jours de données d'accéléromètre porté au poignet, annotées avec deux types d'événements « *onset* », le début du sommeil, et « *wakeup* », la fin du sommeil. Bien que les « *logbooks* », les historiques des événements de sommeil restent la référence, ici c'est le sommeil qui a été désigné comme la plus longue période d'inactivité pendant que la montre est portée. Bien que chaque série soit un enregistrement continu, il peut y avoir des périodes dans la série pendant lesquelles l'accéléromètre a été retiré. Enfin, chaque série de données représente cet enregistrement continu (sur plusieurs jours/événement) pour un sujet expérimental unique. Les données regroupent différentes variables :

- ***series_id*** : un identifiant unique pour chaque série d'accéléromètres
- ***step*** : un pas de temps entier pour chaque observation dans une série
- ***timestamp*** : une date/heure correspondante au format ISO 8601
- ***event*** : le type d'événement, *onset* ou *wakeup*
- ***anglez*** : tel que calculé et décrit par le package GGIR, l'angle z est une métrique dérivée de composants individuels de l'accéléromètre qui est couramment utilisée dans la détection du sommeil et fait référence à l'angle du bras par rapport à l'axe vertical du corps
- ***enmo*** : tel que calculé et décrit par le package GGIR, ENMO est la norme euclidienne moins un de tous les signaux de l'accéléromètre, avec des valeurs négatives arrondies à zéro.

Préprocessing des données

En étudiant les valeurs de l'angle Z (Figure 1A) et de *enmo* (Figure 1B) au cours de la première nuit, nous remarquons que celles-ci fluctuent différemment si l'individu est éveillé ou endormi. Il est donc particulièrement intéressant de se pencher sur l'état de l'individu pendant son sommeil pour mieux comprendre ces variations. Dans le cadre de cette étude, il a été observé que pour un certain nombre d'enfants, les événements liés au sommeil n'ont pas été identifiés. Comme le démontre la figure 2, parmi 277 enfants, 240 présentent des valeurs manquantes en ce qui concerne ces événements. Afin de garantir l'intégrité et la fiabilité de l'apprentissage du réseau de neurones, ces enfants ont été exclus du jeu de données initial. Cette démarche vise à prévenir l'introduction de biais potentiel qui pourrait altérer la qualité des résultats obtenus par le réseau. Ainsi, seules les données de 35 enfants ont été retenues.

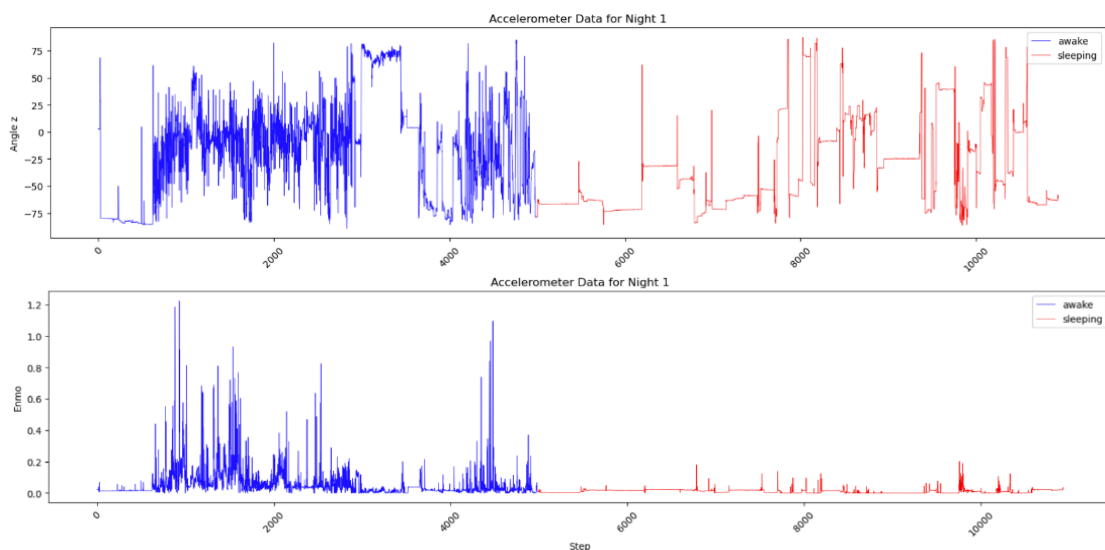


Figure 1 : Valeurs de l'angle Z et de enmo chez un enfant au cours de la première nuit

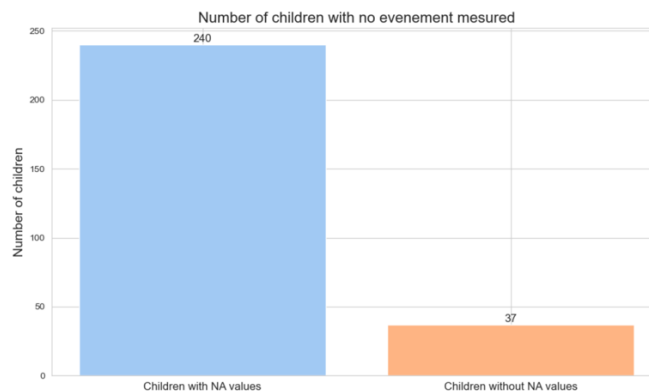


Figure 2 : Graphique représentant le nombre d'enfant comportant des données avec NA ou sans NA

Jeux de données d'apprentissage et de test

Pour la création de nos jeux de données d'entraînements et de tests, nous avons séparé les données selon les individus. Ainsi, le data train d'apprentissage est constitué de 26 individus soit 74,84% des données, et le data test de 11 individus soit 25,16% des données.

Random Forest

Afin d'explorer d'avantage nos données et de mettre en évidence les caractéristiques importantes du jeu de données, une approche préliminaire de prédiction a été réalisé en utilisant la méthode de *Random Forest*. Le *Random Forest* est un algorithme d'apprentissage automatique ensembliste qui combine plusieurs arbres de décision pour effectuer des prédictions. Chaque arbre de décision est construit en utilisant un sous-ensemble aléatoire des données d'entraînement, et les prédictions de chaque arbre sont combinées pour produire un résultat final. Afin d'avoir un modèle optimal, les hyperparamètres choisis sont les suivants : un nombre d'arbres de décision (*n_estimators*) de 500, un nombre minimal d'échantillons requis pour être une feuille dans un arbre de décision (*min_samples_leaf*) à 25, un *random_stage* à 42 pour s'assurer de la reproductibilité des résultats, et un nombre de *n_jobs* à -1 pour accélérer le processus d'entraînement.

Préparation des données d'apprentissage et test

Approche de sliding window

L'approche de fenêtre glissante, souvent dénommée "sliding window", est une technique consacrée dans l'analyse des séries temporelles pour segmenter une séquence prolongée de données en sous-ensembles contigus. Les données sont regroupées par enfants, suggérant que chaque série temporelle représente un enfant distinct. Pour chaque enfant, des fenêtres de taille fixe sont extraites de manière séquentielle, où chaque fenêtre empiète sur la précédente d'un intervalle défini. Ces fenêtres capturées contiennent des informations sur les paramètres 'anglez' et 'enmo', ainsi que leurs étiquettes associées 'event'.

Approche de padding et masking des données

Le *padding* (rembourrage en français) est une approche couramment utilisée pour permettre l'apprentissage de réseaux de neurones notamment en traitement du langage naturel (*NLP*) et dans les réseaux convolutifs (*CNN*). Cette dernière permet de gérer les différences de dimension entre les données d'entrées. Il consiste à ajouter des valeurs (généralement des zéros) aux extrémités des séquences les plus petites afin d'uniformiser la taille des différentes données.

Le *padding* est souvent accompagné d'une autre technique que l'on appelle *masking* permettant de masquer les séquences ajoutées artificiellement afin que le modèle ne les intègre pas au processus d'apprentissage. Le masque généré est souvent de même longueur que le jeu de données obtenu après *padding* et contient des valeurs booléennes indiquant si chaque séquence est valide ou non.

Réseaux neuronaux utilisés

LSTM (Long Short-Term Memory)

Le *Long Short-Term Memory (LSTM)* est un réseau de neurones cherchant à résoudre le problème de la disparition du gradient car celui-ci peut dans certains cas empêcher l'altération des poids durant l'apprentissage à cause de valeurs de gradient trop petites. C'est un outil puissant qui permet de prédire des données basées sur des données chronologiques ce qui en fait un choix privilégié pour l'analyse.

Réseau de Neurones Convolutif (CNN)

Un Réseau de Neurones Convolutif (CNN) est conçu pour apprendre automatiquement et de manière adaptative des hiérarchies spatiales de caractéristiques à partir des données d'entrée. Les CNN se composent de plusieurs couches, notamment des couches de convolution, des couches de regroupement et des couches entièrement connectées. Les couches de convolution appliquent des filtres aux données d'entrée pour détecter des caractéristiques et des motifs, et les couches de regroupement réduisent les dimensions spatiales. L'optimiseur Adam (pour *Adaptive Moment Estimation*) quant à lui est un algorithme d'optimisation couramment utilisé pour mettre à jour les poids et les biais des couches de réseau neuronal pendant l'entraînement. Il s'agit d'une variante de la méthode d'optimisation de la descente de gradient stochastique (SGD). L'idée est d'adapter les taux d'apprentissage pour chaque paramètre individuellement en fonction des gradients passés et de l'historique cumulatif des gradients. Le modèle construit commence par des couches de convolution 1D pour extraire des caractéristiques à partir des données en entrée. Deux couches de convolution sont empilées, chacune avec 64 filtres, un noyau de taille 3 et une fonction d'activation ReLU pour introduire une non-linéarité. Les couches de max-pooling sont utilisées pour réduire les dimensions spatiales des caractéristiques extraites. Ensuite, une couche de *flattening* transforme les sorties des couches de convolution en un vecteur plat, ce qui prépare les données pour la couche dense suivante. La couche dense a 64 neurones avec une activation ReLU, et une couche de dropout réduit le risque de surapprentissage en désactivant aléatoirement 50 % des neurones pendant l'entraînement. Enfin, la couche de sortie utilise une fonction d'activation sigmoid pour la classification binaire. L'optimiseur Adam est configuré avec un faible taux d'apprentissage de 0.00001 pour mettre à jour les poids du modèle, et la perte de cross-entropie binaire est utilisée pour évaluer l'erreur.

GRU

Les GRU sont une autre variante de réseau neuronal récurrent (RNN) conçue pour résoudre certains des problèmes de convergence que rencontrent les LSTM. Les GRU sont plus simples en termes de structure que les LSTM, ce qui les rend plus faciles à former et moins susceptibles de souffrir de surajustement dans certaines situations.

Résultats

Traitement de données

Dans l'analyse des données prétraitées, il est apparu que le nombre d'événements "awake" (éveillé) est nettement supérieur à celui des événements "sleeping" (endormi), avec respectivement 8 660 676 et 4 504 884 occurrences (Figure 3). Cette disparité reflète une réalité comportementale où un individu passe généralement plus de temps à l'état éveillé qu'à l'état endormi. Toutefois, face à cette inégalité des classes, il pourrait être tentant d'équilibrer les données en éliminant certains événements "awake". Néanmoins, il est essentiel de conserver l'intégralité de ces événements pour plusieurs raisons. D'une part, supprimer des données "awake" pourrait introduire un biais dans le modèle et altérer sa capacité à reconnaître correctement cet état. D'autre part, la représentation authentique de la répartition des états offre une meilleure généralisation du modèle de deep learning dans des conditions réelles d'utilisation. De plus, cette disparité peut offrir des insights précieux sur le comportement et les habitudes de sommeil des individus, permettant ainsi de mieux comprendre et prédire leurs tendances en matière de sommeil.

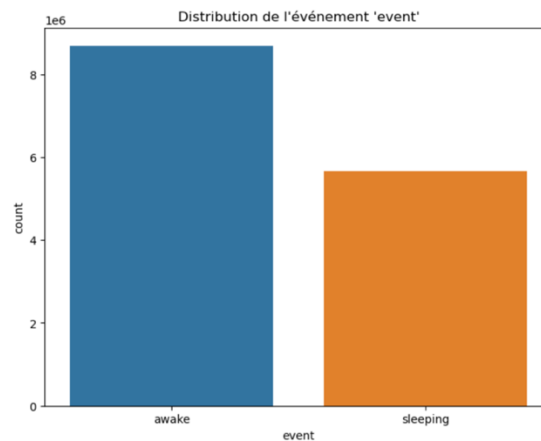


Figure 3 : Distribution des évènements de sommeil dans le jeu de données après préprocessing.

Suite à cela, une analyse de la distribution des valeurs "Enmo" et "Anglez" a été effectuée pour déterminer s'il existe des différences significatives entre les classes « awake » et « sleeping ». La figure 4 illustre les valeurs moyennes et médianes pour ces mesures. Cette analyse montre des variations marquées en fonction de l'état observé. Pour "Enmo", l'état éveillé présente des valeurs distinctement plus élevées, tant en moyenne qu'en médiane, par rapport à l'état endormi. Concernant "Anglez", bien que l'écart entre la moyenne et la médiane soit modeste à l'état éveillé, à l'état endormi, la moyenne se situe nettement en dessous de la médiane. Ces observations indiquent une divergence dans la distribution des données selon l'état de l'enfant, suggérant que la classification de l'état pourrait être réalisable en se basant sur les mesures "Enmo" et "Anglez".

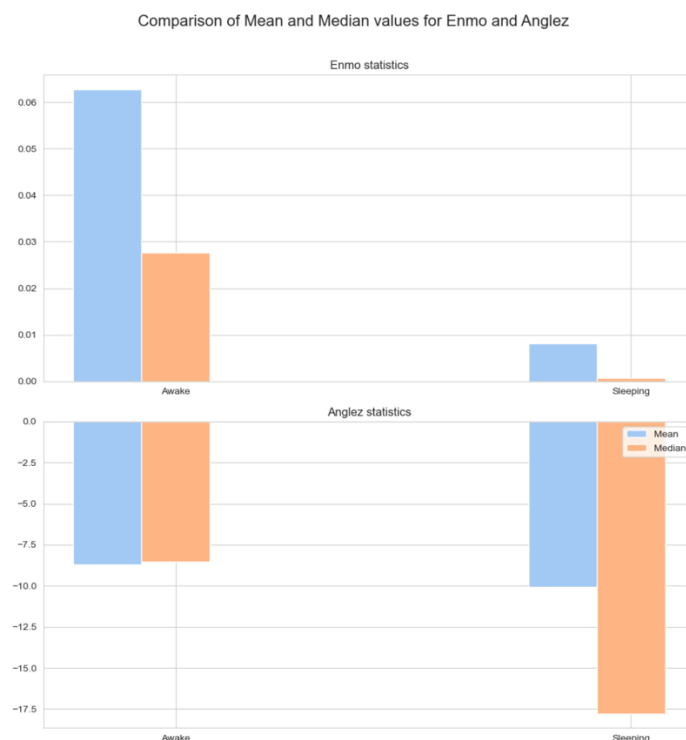


Figure 4 : Graphique représentant la moyenne et la médiane des valeurs de Enmo et d'angleZ selon les événements Awake et Sleeping.

La technique du Kmeans a par la suite été utilisée afin d'identifier s'il est possible de classifier les événements awake et sleeping à partir des données d'enmo et anglez. La figure 5A illustre la distribution des événements de sommeil. Ce dernier montre une concentration élevée d'événements "awake" dans les régions à faible enmo et anglez. La figure 5B montre les résultats de la classification

K-means, où deux clusters distincts se forment, suggérant une séparation potentielle des états "awake" et "sleeping". Cependant, la matrice de confusion (Figure 5C) révèle que bien que le K-means puisse identifier certains patterns, il y a une superposition significative, indiquant une précision de classification moins que parfaite.

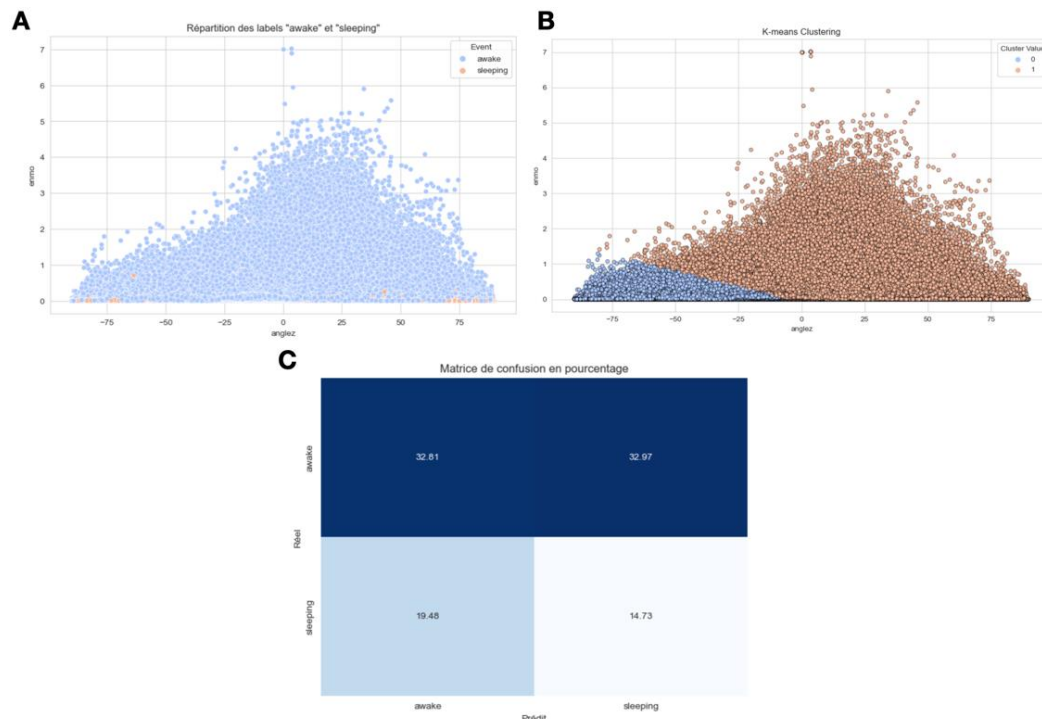


Figure 5 : *A. Observation de la répartition des classes en fonction d'enmo et anglez. B. Résultats de la classification par la méthode des K-means. C. Matrice de confusion de la classification obtenue à l'aide de la méthode des K-means.*

Méthodes d'apprentissage automatique

Random Forest

Dans le cadre de notre étude, nous avons d'abord adopté une approche basée sur la méthode de Random Forest pour la classification. La figure 6A que le paramètre "anglez" est plus important pour le modèle que "enmo". En effet, "anglez" possède une importance proche de 0,6 tandis que "enmo" se situe autour de 0,4. Concernant la performance du modèle (figure 5B), la courbe ROC affiche une aire sous la courbe (AUC) de 0,75, ce qui est considéré comme une performance acceptable, mais pas optimale.

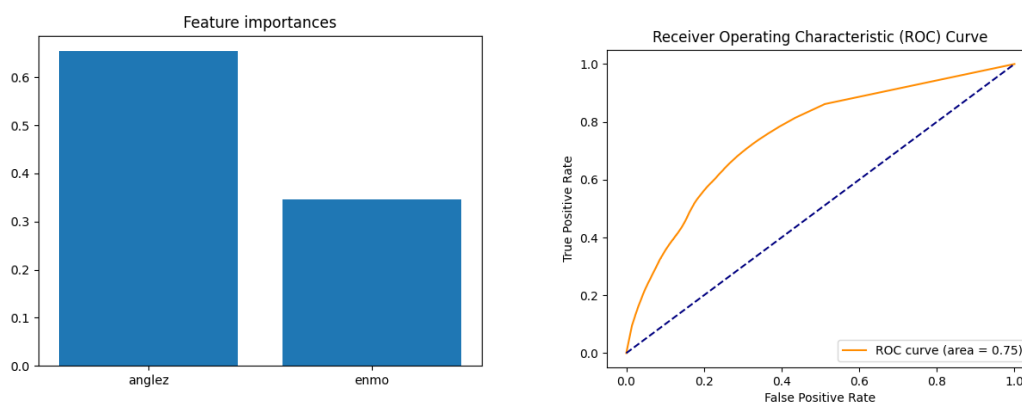


Figure 6 : (A) Représente la contribution des variables anglez et enmo pour la prédiction du modèle. (B) Courbe ROC du modèle Random Forest

Architectures de réseaux de neurones

Performances obtenues par l'approche "sliding window"

LSTM

Après avoir évalué la performance de notre modèle basé sur le Random Forest, nous avons décidé d'explorer une approche basée sur les réseaux de neurones, plus précisément avec le modèle LSTM (Long Short-Term Memory). Les LSTMs sont particulièrement adaptés pour les séries temporelles, car ils peuvent capturer des dépendances à long terme. Les résultats présentés sur la figure 7 montrent une diminution progressive de la loss sur les données d'entraînement et de validation jusqu'à atteindre une valeur de 0.25. Concernant la précision du modèle, ce dernier augmente progressivement jusqu'à atteindre environ 90% de bonnes prédictions. La matrice de confusion montre que 10,51% de mauvaises prédictions et un score F1 égal à 0.9207. Ainsi, les performances du modèle semblent être correctes. Toutefois, malgré les performances respectables du modèle LSTM, celui-ci a ses limites. Les LSTMs peuvent être gourmands en ressources et nécessitent une grande quantité de données pour être entraînés efficacement. En effet, le modèle nécessite un batch size de 256.

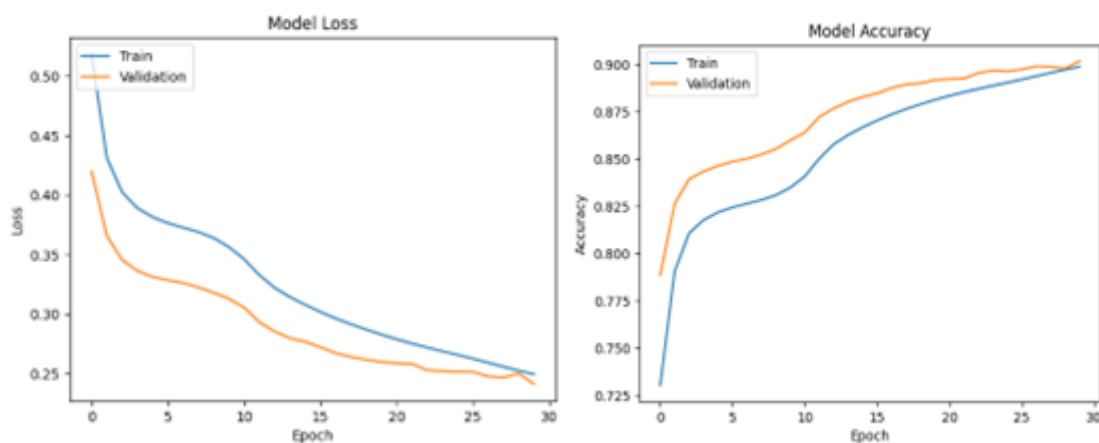


Figure 7 : Graphique représentant : **A.** la perte du modèle LSTM. **B.** la performance du modèle.

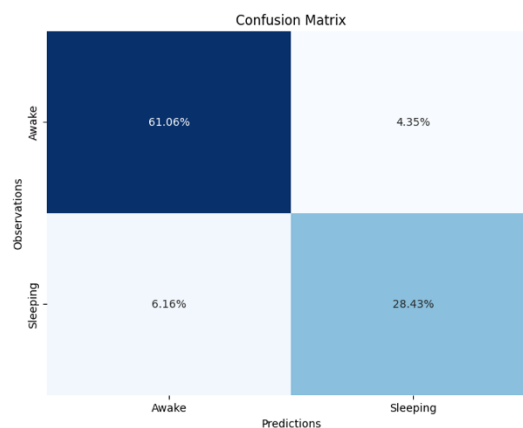


Figure 7 C: Matrice de confusion du modèle LSTM

Compte tenu de ces limites et après une évaluation approfondie, nous avons décidé d'opter pour un modèle GRU (Gated Recurrent Unit). Le GRU, bien que similaire au LSTM dans sa capacité à traiter les séries temporelles, est souvent plus efficace en termes de temps de calcul et nécessite moins de paramètres. Dans notre contexte, le GRU s'est avéré être une meilleure option pour optimiser la performance tout en réduisant la complexité du modèle. La structure du réseau est représentée sur la figure 8. Chaque couche comporte 64 neurones. Le modèle a un dropout de 0.2 et 13121 paramètres.

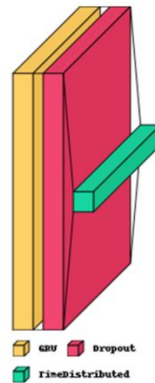


Figure 8 : Représentation de la structure du GRU

Le modèle GRU (Gated Recurrent Unit) a montré une efficacité notable dans le traitement des séries temporelles. D'une part, il présente une convergence rapide, comme le montrent les graphiques de "loss" et de précision. En effet, la courbe de "loss" de validation montre une stabilisation autour de 40-50 époques, indiquant une adaptation optimale du modèle (Figure 9). Par ailleurs, la précision de validation atteint presque 94%, ce qui est un indicateur de sa robustesse. Toutefois, même si seulement 7,47% des données sont mal prédites, cela suggère qu'il reste une marge d'amélioration, surtout lorsqu'on analyse la matrice de confusion disponible sur la figure 10, où des erreurs de classification pour les états "awake" et "sleeping" sont encore présentes.

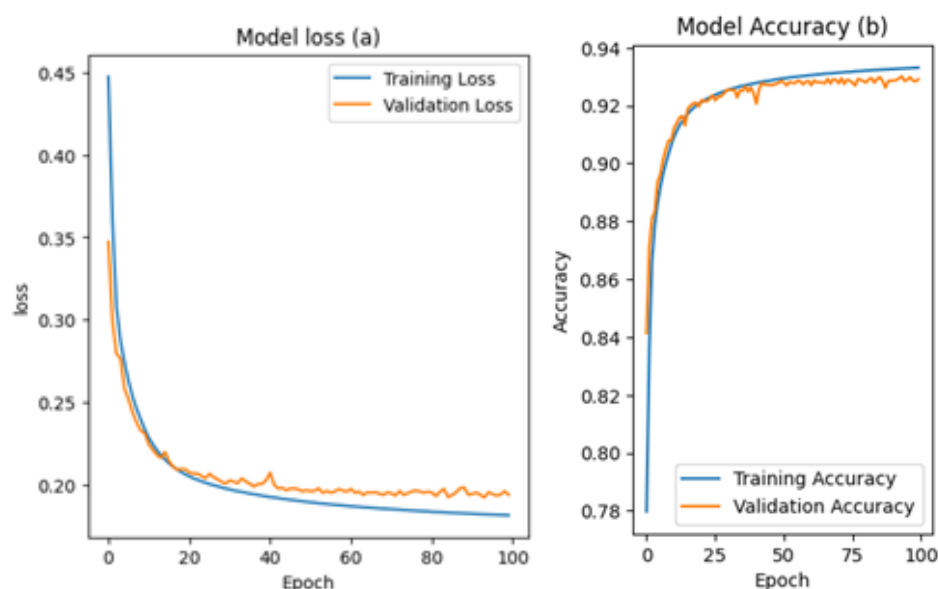


Figure 9 : Graphique représentant : **A.** la perte du modèle. **B.** ses performances.

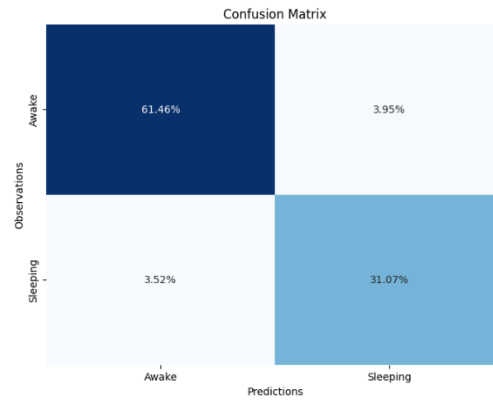


Figure 10 : *Matrice de confusion du modèle GRU*

Lorsque l'on s'intéresse aux erreurs de classification, il est possible de voir sur la figure 11 que 7 enfants ont des erreurs de classification et que ces erreurs concernent principalement l'évènement awake. Cela s'explique car le nombre d'évènement awake est nettement supérieur à celui de sleeping. Si l'on s'intéresse aux erreurs de classification selon le jour (Figure 12), nous voyons que le jour ayant le plus d'erreurs est le mardi. Cependant, nous nous attendions à ce que les mauvaises prédictions soient le week-end car comme nous pouvons le voir sur la figure 13, les enfants dorment plus tard à ce moment-là. Cela montre que le modèle réussit à identifier que les enfants dorment plus tard le week-end.

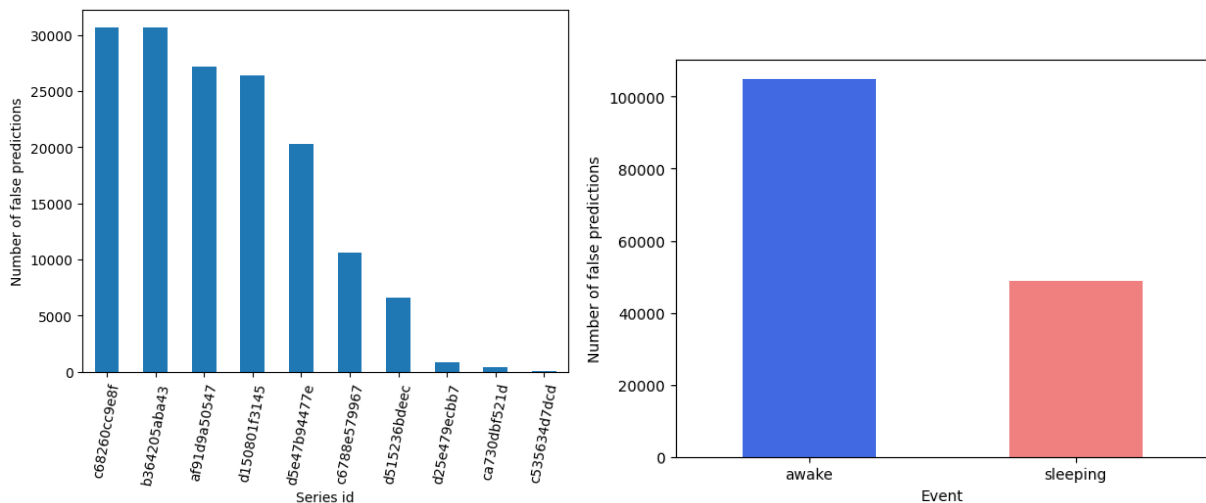


Figure 11 : *Nombre d'erreurs de prédiction (A) par individus (B) par évènements*

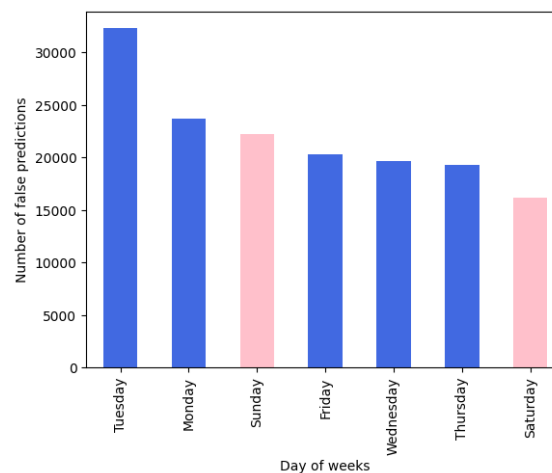


Figure 12 : Nombre d'erreurs de prédiction selon le jour de la semaine

Interprétation des résultats obtenus

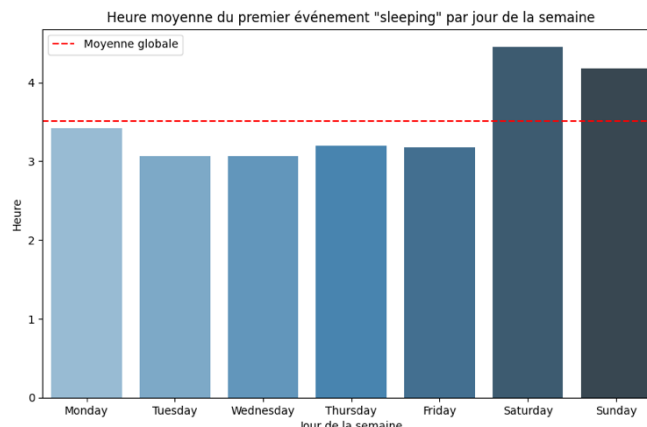


Figure 13 : Heure moyenne du premier événement "sleeping" par jour de la semaine

Afin de comparer ce modèle avec une autre architecture de neurones, un modèle de type Réseaux Neuronaux Convolutifs (CNN) a été utilisé puisque ce type de réseau peut être utilisé pour le traitement des séries temporelles et la détection de motifs dans des signaux complexes tels que ceux de enmo et anglez. Contrairement aux modèles récurrents, les CNN peuvent automatiquement et adaptativement apprendre des motifs spatiaux à partir de données d'entrée, ce qui est idéal pour le traitement de signaux et de séries temporelles. En utilisant des couches convolutives, les CNN peuvent capter des caractéristiques hiérarchiques des données, ce qui pourrait être particulièrement utile pour distinguer entre les états "awake" et "sleeping" dans des signaux complexes.

Performances obtenues par l'approche "padding-masking"

L'une des limites auxquelles fait face l'approche par fenêtre glissante est la perte de l'information liée à l'individu (*en d'autres termes la série temporelle*) traitée au moment T. En effet, le découpage en plusieurs fenêtres crée des chevauchements entre les observations de séries adjacentes et ne permet pas au modèle de cerner à quel patient appartient chaque observation. Cela peut avoir comme effet de biaiser le modèle par la détection de relations ou corrélations inexistantes entre séries voisines ou par l'omission de certaines relations intra-séries de par la discontinuité de leurs représentations.

Pour tenter d'apporter une vision différente, nous proposons une seconde approche complémentaire dont le parti pris est de garder la segmentation originelle des données. Chaque individu (*enfant*) est décrit par une série temporelle (*ensemble d'observations*) de taille différente. Afin de permettre l'apprentissage des architectures de réseaux de neurones que nous avons définies, nous avons ajouté des observations artificielles (*padding*) aux individus qui en contenaient le moins. Ces observations sont définies par 3 variables : anglez, enmo et step dont les valeurs sont nulles (*zero-padding*).

Les observations ajoutées sont également masquées pour le modèle à l'aide d'une couche "Masking" afin d'éviter qu'elles soient considérées au cours de l'apprentissage. Nous testons 4 différents modèles pour cette approche : *LSTM*, *BiLSTM*, *CNN-LSTM* et *CNN-BiLSTM*. Nous voulons ainsi déterminer à la fois si l'utilisation de réseaux bidirectionnels peut avoir une incidence

sur nos métriques de classification, mais également si l'extraction de caractéristiques à l'aide d'un *CNN* offre un avantage par rapport à l'utilisation d'une architecture récurrente seule.

Le nombre d'observations totales par classes étant déséquilibré à raison de 4,504,884 observations pour la classe *onset* et de 8,660,676 pour la classe *asleep* nous optons pour l'utilisation de métriques robustes aux déséquilibres de classes telles que l'AUC et le score F1.

Phase d'apprentissage

Durant la phase d'apprentissage, les quatre modèles démontrent une évolution cohérente de la fonction de coût sur l'ensemble d'apprentissage, ce qui témoigne de l'optimisation de cette dernière et du bon déroulement de l'apprentissage.

Cependant, l'évolution de la fonction coût sur l'ensemble de validation vient y apporter certaines nuances. En effet, les modèles *LSTM* et *BiLSTM* semblent correctement généraliser les relations apprises sur de nouvelles données ce qui témoigne de l'absence de sur-apprentissage sur ces architectures, tandis que les modèles utilisant un *CNN* éprouvent plus de difficultés. L'architecture *CNN-LSTM* arrive à minimiser quelque peu la *loss* sur l'ensemble de validation néanmoins, sa valeur reste nettement plus élevée que celle d'apprentissage, tandis que pour le *CNN-BiLSTM*, la fonction se met même à croître. Cela peut être un indicateur de sur-adaptation aux données qui serait notamment dû au fait que ces modèles soient trop sophistiqués pour les données qu'ils représentent ce qui les empêche de généraliser sur de nouvelles observations.

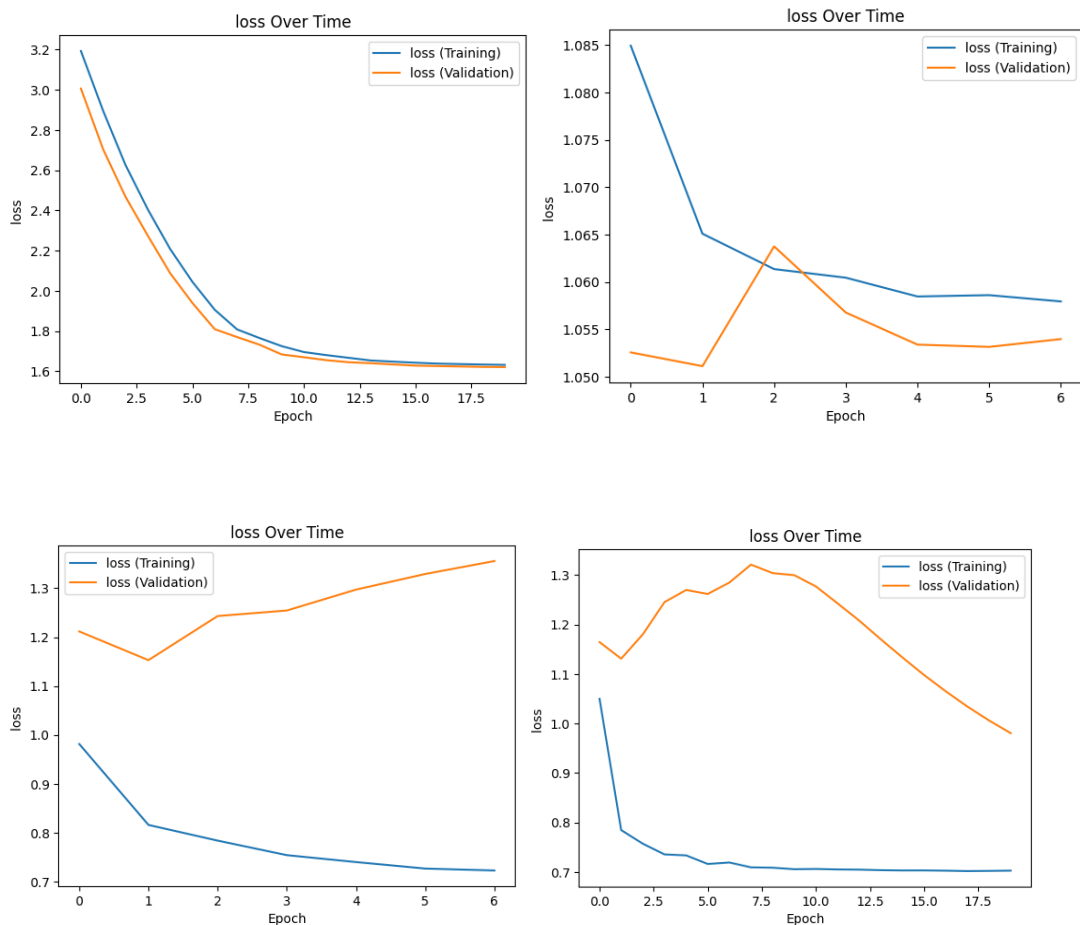


Figure 14 : Evolution des fonctions de coût sur l'ensemble d'apprentissage et de validation. LSTM (haut-gauche), BiLSTM (haut-droite), CNN-BiLSTM (bas-gauche), CNN-LSTM (bas-droite)

Phase d'évaluation (test)

Lorsque l'on se penche sur l'évaluation des modèles par les métriques de classification sur le l'ensemble de test, comme le montre le tableau 1 ci-dessous, nous pouvons remarquer plusieurs points :

- Malgré la minimisation non-optimale de sa fonction coût sur l'ensemble de validation, le CNN-LSTM semble obtenir les meilleures performances en AUC et F1-Score.
- Le LSTM et BiLSTM ne présentent pas de différences majeures de performances, l'aspect bidirectionnel ne semble donc pas avoir d'impact significatif sur les performances.
- L'utilisation d'un CNN pour l'extraction des caractéristiques peut effectivement avoir un intérêt mais elle peut également trop complexifier le modèle pour des données triviales.

Tableau 1 : Comparatif des performances des différents modèles utilisant l'approche padding-masking.

Modèle	Précision	Rappel	F1-Score	AUC	Accuracy
LSTM	0.30	0.37	0.29	0.59	0.37
BiLSTM	0.45	0.42	0.26	0.59	0.42
CNN-LSTM	0.53	0.68	0.60	0.86	0.68
CNN-BiLSTM	0.42	0.44	0.30	0.52	0.44

Conclusion

Suite à une exploration rigoureuse des architectures de réseaux de neurones pour le traitement des séries temporelles, nous avons constaté que le modèle GRU, en particulier, offre des performances significatives dans la classification des états de sommeil. Le modèle que nous avons utilisé est simple et permet d'assez bien classifier l'état de sommeil des enfants. La rapidité de convergence, combinée à une précision notablement élevée, démontre l'adaptabilité et la robustesse du modèle face aux complexités intrinsèques des données de sommeil. Le sommeil, en tant que phénomène biologique, est essentiel pour le bien-être physique et mental. Dans le contexte des enfants, une analyse approfondie du sommeil est d'autant plus cruciale. Les motifs de sommeil durant l'enfance ont un impact direct sur le développement cognitif, émotionnel et physique. Des perturbations ou des anomalies dans les schémas de sommeil peuvent avoir des conséquences à long terme sur la santé mentale et physique d'un enfant. La nécessité d'analyser le sommeil chez les enfants provient non seulement de la volonté de comprendre leurs besoins en matière de repos mais aussi de l'importance de détecter précocement des troubles potentiels. Des conditions comme l'apnée du sommeil, les terreurs nocturnes ou même des problèmes de sommeil liés à des troubles neurologiques peuvent être identifiées plus tôt grâce à de telles analyses, permettant ainsi des interventions plus précoces et plus efficaces. En fin de compte, alors que nous avons réalisé des progrès significatifs dans la classification des états de sommeil grâce à des modèles comme le GRU, la voie est ouverte pour des recherches plus approfondies. L'intégration d'approches plus sophistiquées, la combinaison de modèles et l'utilisation de données supplémentaires peuvent nous aider à peaufiner davantage nos méthodes, avec l'objectif ultime d'améliorer la santé et le bien-être des enfants.

Bibliographie

Sundararajan, Kalaivani et al. "Sleep classification from wrist-worn accelerometer data using random forests." *Scientific reports* vol. 11,1 24. 8 Jan. 2021, doi:10.1038/s41598-020-79217-x.

Nunavath, Vimala et al. "Deep Learning for Classifying Physical Activities from Accelerometer Data." *Sensors (Basel, Switzerland)* vol. 21,16 5564. 18 Aug. 2021, doi:10.3390/s21165564.

Cho, Taeheum, et al. "Deep-ACTINet: End-to-End Deep Learning Architecture for Automatic Sleep-Wake Detection Using Wrist Actigraphy." *Electronics*, vol. 8, no. 12, Dec. 2019, p. 1461. *Crossref*, doi:10.3390/electronics8121461.

Chen, Zhenghua et al. "An Attention Based CNN-LSTM Approach for Sleep-Wake Detection With Heterogeneous Sensors." *IEEE journal of biomedical and health informatics* vol. 25,9 (2021): 3270-3277, doi:10.1109/JBHI.2020.3006145.

ANNEXE

Fichiers utilisés

train_series.parquet* et *test_series.parquet : Séries respectivement utilisée comme données d'entraînement et de test où chaque série est un enregistrement continu de données d'accéléromètre pour un seul sujet sur plusieurs jours contenant les champs suivants (voir Tableau 1) :

- *series_id* : un identifiant unique pour chaque série d'accéléromètres
- *step* : un pas de temps entier pour chaque observation dans une série
- *timestamp* : une date/heure correspondante au format ISO 8601 %Y-%m-%dT%H:%M:%S%z
- *anglez* : tel que calculé et décrit par le package GGIR, l'angle z est une métrique dérivée de composants individuels de l'accéléromètre qui est couramment utilisée dans la détection du sommeil et fait référence à l'angle du bras par rapport à l'axe vertical du corps
- *enmo* : tel que calculé et décrit par le package GGIR, ENMO est la norme euclidienne moins un de tous les signaux de l'accéléromètre, avec des valeurs négatives arrondies à zéro

	series_id	step	timestamp	anglez	enmo
0	038441c925bb	0	2018-08-14T15:30:00-0400	2.636700	0.0217
1	038441c925bb	1	2018-08-14T15:30:05-0400	2.636800	0.0215
2	038441c925bb	2	2018-08-14T15:30:10-0400	2.637000	0.0216

train_events.csv : Historique de sommeil pour les séries du jeu de données d'entraînement enregistrant les événements de début et de réveil contenant les champs suivants (voir Tableau 2) :

- *series_id* : identifiant unique pour chaque série de données d'accéléromètre dans *train_series.parquet*.
- *night* : une énumération de paires d'événements d'*onset/wakeup* potentiels. Au maximum une paire d'événements peut se produire chaque nuit.
- *event* : le type d'événement, qu'il s'agisse de son apparition ou de son réveil.
- *step* et *timestamp* : l'heure enregistrée de l'événement dans la série d'accéléromètres.

	series_id	night	event	step	timestamp
0	038441c925bb	1	onset	4992.0	2018-08-14T22:26:00-0400
1	038441c925bb	1	wakeup	10932.0	2018-08-15T06:41:00-0400
2	038441c925bb	2	onset	20244.0	2018-08-15T19:37:00-0400