TRANSPARENT SLEEP APNEA DETECTION: A COMPARATIVE STUDY OF XAI METHODS ON 1D-CNNS

Aron Lapp
Affiliation
HTWG Konstanz
Aron.Lapp@htwg-konstanz.de

ABSTRACT

Non-invasive sensors for detecting sleep-related breathing disorders offer great potential for making sleep diagnostics more accessible and efficient. This paper explores how Explainable AI (XAI) methods can enhance the interpretability of Deep Learning models trained on sleep-related sensor data.
A 1D-Convolutional Neural Network (1D-CNN) was trained on the SHHS2 dataset using four biosignals to detect apnea events. The model achieved high performance (ROC AUC (Receiver operating characteristic area under the curve): 0.949, accuracy: 89.4 %). To explain the model's key drivers for decision making, four XAI techniques, Grad-CAM, LIME, Deep SHAP and Integrated Gradients were implemented and systematically compared based on faithfulness, sparsity and runtime.

***K*eywords** Sleep Stage Classification · Explainable AI · Model Interpretability · Deep Learning; Polysomnography · Physiological Signal Processing · Transparency in AI · Human-AI Interaction

1 Introduction

In modern medicine, the importance of healthy sleep is increasingly recognized. With its growing popularity, there is high demand for the collection and evaluation of sleep-related data. Traditionally, this data is acquired via overnight polysomnography in a dedicated sleep laboratory. This means patients have to spend a night wired to specialized sensors and generate data which is monitored by clinicians.
Though this process is highly informative, it is also both expensive, due to the price of hardware and staff, and inefficient since each subject requires a dedicated setup to collect sleep data.

To address these limitations, recent efforts focused on developing processes in which subjects could use more minimal, non-invasive sensors to track their own sleep in either more simple, less costly hospital-beds or even at home. However, the data collected this way still requires staff to analyze.

The recent rise of Deep Learning applications offers a promising alternative approach to detect sleep related events (eg. apnea, hypopnea) in the collected sleep data. Yet deep neural networks are naturally opaque. They are extremely complex and hard to grasp for a human brain which makes it hard to trust them completely. Especially in a medical context, it is absolutely mandatory to be certain about diagnosis and any decisions made about the subject's health. To gain better insights on how a neural network makes its decisions, special methods to gain insights on the decision process of a neural network, called Explainable AI (XAI) have been developed recently. This paper focuses on exploring how XAI methods can be integrated with Deep Learning based sleep analysis to deliver both high diagnostic accuracy and the transparency required for clinical adoption. Several state-of-the-art XAI techniques will be evaluated and compared, assessing their ability to actually explain the model's decisions without impacting performance. This could help further research decide what kind of XAI works best for a specific use case. In a longer scope, functional, well researched XAI could help automate various clinical analyses while avoiding drawbacks on trust and security.

2 Dataset and preprocessing

Data were obtained from the national Sleep Research Resource (NSRR) via Sleepdata.org. To train the model, all 2651 patient's data from SHHS2 was used. For evaluation and explainable AI methods, a subset of 100 patients from SHHS1 was used.

2.1 Signal selection and segmentation

The biosignals oxygen saturation (SpO_2), pulse rate (HR), thoracic respiratory effort (Thor-Res) and abdominal respiratory effort (Abdo-Res) were extracted from all SHHS2 EDF recordings. All biosignals were downsampled to 1Hz across all channels. Furthermore, reducing signal density leads to faster training times as less memory overhead. Kristiansen et al. [1] chose 1Hz before and did not report any problems. However it is possible that some rapid SpO_2 desaturations or any other fast changes to one of the biosignals might be lost. Recordings were then divided into overlapping 60-second windows with a stride of 30 seconds, in line with AASM guidelines to capture apnea / hypopnea chunks lasting at least 10 seconds within a window.

2.2 Preprocessing

As the model should be capable of detecting sleep apnea from real sensor data, the preprocessing was kept as minimal as possible. As SHHS datasets grew over time, the header naming for biosignals is not uniform. Therefore as a first step, all headers were normalized. The normalized data was then loaded using MNE, the four channels were picked and resampled to the lowest of the 4 signal frequencies (1Hz) using anti-aliasing and FIR filters. After that, each 60-second window was extracted with a stride of 30 seconds in order to not lose apnea events that were cut off by the windowing. Over every 60-second window, all values were normalized using l2 norm channel-wise. XML annotations from SHHS2 were parsed to identify chunks with obstructive, central or mixed apneas or hypopneas with a duration of at least 10 seconds. Any window containing an uninterrupted apnea or hypopnea event of at least 10 seconds was assigned label 1, all other were assigned label 0.

3 Model and training

The model architecture is inspired by the best model found by Alarcón et al. [2], as it is established the model is working well in both prediction and explainability, especially regarding explainability using Grad-CAM which requires image-generating models.

The model used in this work is simpler than the best one found by Alarcón et al. [2] since this work's focus is not the classification itself but the explainability of the trained model. Findings regarding explainability researched using simpler models can be applied to more complex models.

The structure used consists of the following and is also shown in Figure 1:

- **Input Layer** expects sequences of length 60 with 4 parallel channels
- **First Convolutional Block** is a Conv1D block with 64 filters of width 3. It keeps the sequence length at 60 and extracts 64 feature maps. It uses ReLU activation to induce nonlinearity and finishes with a Max-Pooling1D block with pool size 2 to yield a 30-second signal to the next block.
- **Second Convolutional Block** is another Conv1D block with 128 filters of width 3, which extracts more complex features from the 30 second signal it was passed. The second Conv1D block again uses ReLU activation and another Max-Pooling1D block to fold the signal to 15 seconds.
- **Third Convolutional Block** is a Conv1D block with 256 filters using ReLU again and finishes with GlobalAveragePooling1D to collapse each of the 256 feature maps to a single value producing a vector of dimension (256×1).
- **Fully Connected Layers** start with a Dense(128) block with ReLU activation to learn the higher level combinations of the 256 features. The dropout rate was set to 0.3.
- **Output Layer** is a dense(1) block with sigmoid activation to produce single probabilities between 0 and 1 for binary classification

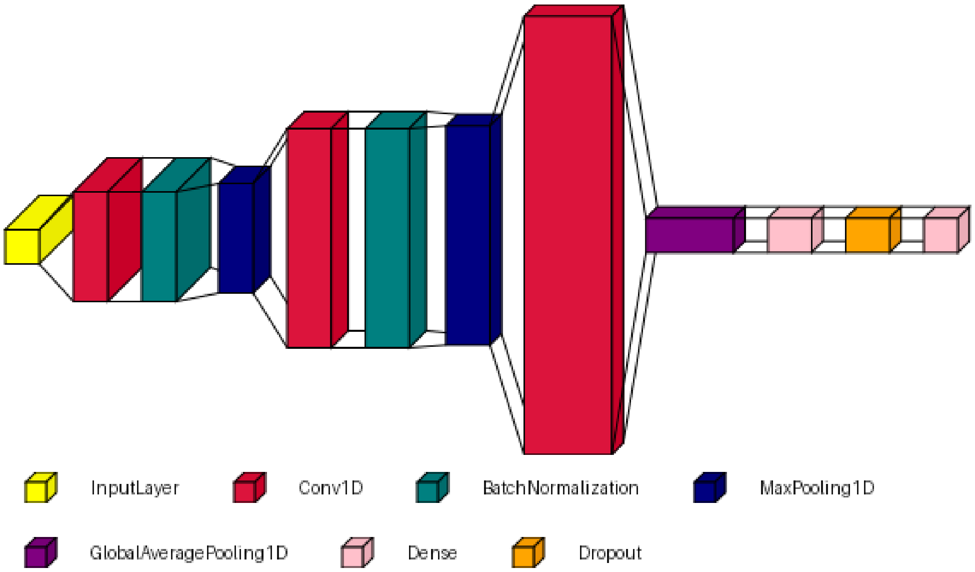


Figure 1: Architecture of neural network used for classifying apnea events on 60 second windows

Table 1: Training hyperparameters

Parameter	**Value**	**Notes**
Optimizer	Adam (LR = 1×10^{-3})	–
Batch size	128	Chosen for GPU memory
Epochs (max)	50	Early-stop after 5 epochs w/o val. gain
Early-stopping patience	5 epochs	Monitors validation ROC-AUC
Dropout rate	0.30	After Dense(128)
Weight init.	He normal	For all Conv1D layers

The first two convolutional layers additionally use batch normalization to speed up training. As the optimizer, Adam was chosen with a learning rate of 1×10^{-3}. As the classification problem is binary, binary cross-entropy is the loss-function of choice. The training metrics were:

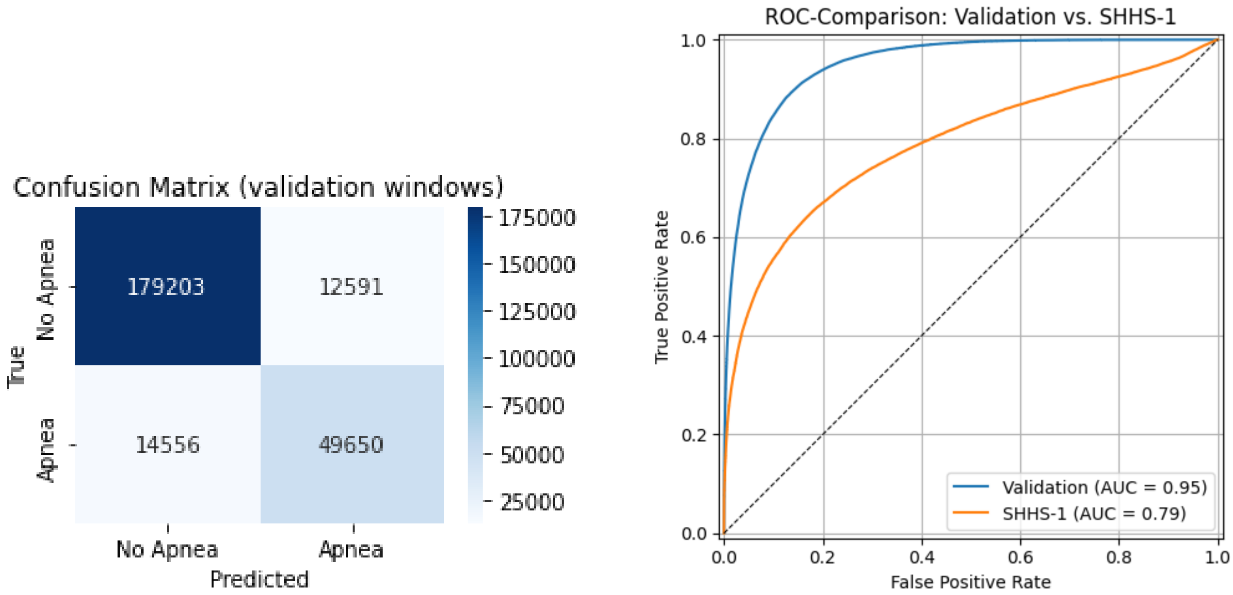
- **ROC-AUC**, which is a numeric indicator for ROC
- **Precision**, which is a measure for what fraction of samples labeled as a class are actually part of that class
- **Recall**, which is a measure for out of all members of a label, what fraction was labeled with that exact label

Table 1 shows the hyperparameters chosen for training.

Data preprocessing and model training were performed in a jupyter notebook using Python 3.12, Keras 3.6.0, NumPy 1.26.4, Tensorflow 2.18.0 and Scikit-Learn 1.7.0. The workstation used for preprocessing and training contained an AMD EPYC 7J13 processing unit, 216 GB of RAM and a NVIDIA A100-SXM4-40GB GPU. One full training iteration took ca. 90 minutes. 80 % of the available data was used for training and 20 % for validation.

3.1 Evaluation

On the validation set (256000 windows), the network correctly classified 89.4 % of all 60-second segments. Its ROC-AUC is 0.949, indicating that any randomly chosen apnea window receives a higher activation than any randomly chosen non-apnea window in 94.9 % of cases. Under the precision recall curve, the model achieves an AUC of 0.871, which means that it is robust even though only about 25 % of the windows in the data set contain apnea events.
Breaking performance down by class shows complementary strengths: for non-apnea segments (labeled 0), precision is 0.925 and recall is 0.934. This means that the model almost never mistakes a non-apnea window for an apnea window while detecting 93.4 % of non-apnea windows as such. For apnea segments (labeled 1), precision is 0.798 and recall is



(a) Confusion Matrix of the trained model

(b) ROC curve of the trained model

Figure 2: (a) Confusion Matrix and (b) ROC curve of the trained model.

0.773. This means almost 80 % of the windows the model detects as label 1 are actually label 1 and 77.3 % of events with real apnea are detected as such.
Finally, the gap between training and test metrics is very small (training ROC-AUC is 0.954, testing ROC-AUC is 0.949), indicating minimal overfitting. This means the model is not overfitted and only has a small generalization gap, which means the model would also do well when presented data that were not included in training data.
The corresponding confusion matrix shows that the majority of the predictions are correct.
The ROC curve shown in Figure 2b shows the ROC Curve of the validation data colored blue and the ROC Curve of test data from SHHS1. It is clearly visible that the model performs significantly worse on a held out test dataset but is still far from guessing. Therefore XAI methods can still be applied using test data from SHHS1.

4 Explainability

In medical applications such as sleep apnea detection, model accuracy alone is not safe enough. Both clinicians and patients need to understand why a model made a certain decision. Explainable AI (XAI) methods aim to open the 'black box' of deep neural networks by attributing predictions to human-interpretable features. In the context of 1D biosignals, we focus on three metrics [4]

1. Faithfulness: Explanations should faithfully reflect the model's internal reasoning
2. Sparsity: Only a small subset of time points or channels should be highlighted to simplify interpretation
3. Runtime: Computation should be as fast as possible

This work compares four state of the art XAI methods:

Grad-CAM for 1D Signals
Gradient-weighted Class Activation Mapping (Grad-CAM) [6] was originally developed for images. The trained neural network adapts it to 1D by:

- Computing the gradient of the predicted class score with respect to each channel of the final convolutional activation (shaped 60×256)
- Averaging those gradients over time to obtain a single weight per feature map

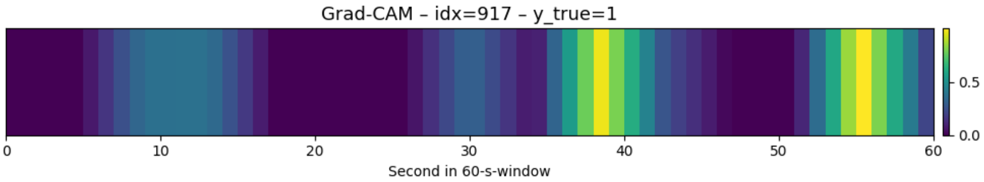


Figure 3: Grad-CAM activation map for a single 60-second window

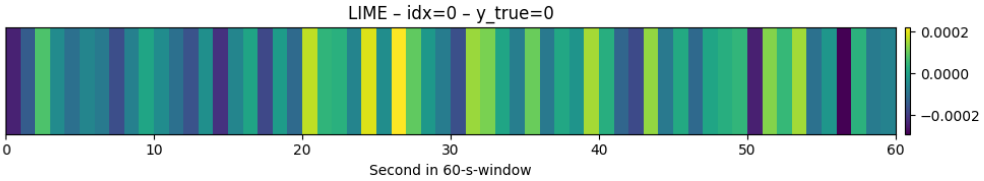


Figure 4: LIME activation map for a single 60-second window

- Forming a weighted sum of the feature maps applying ReLU to remove negatives and only show sections with positive impact
- Upsampling back to the 60-second window

The resulting heatmap highlights the time points and channels that were most significant for the decision. Figure 3 shows a Grad-CAM map of one 60-second time slice. The regions marked yellow have high activation. The data was labeled 1 in SHHS2 annotations. The neural network highlights two areas that have a combined duration over 10 seconds and therefore labels the slice with W1.

LIME

Local Interpretable Model-agnostic Explanations (LIME) [7] approximate the model's behavior in the neighborhood of one example by fitting a sparse linear vector. The following steps were applied to use LIME with the trained model:

- Randomly mask or jitter small segments of the 60-second signal to generate synthetic samples
- Fit a weighted linear model on the outcome using the network's outputs as targets
- Interpret the vector's coefficients to rank which 5-second chunks contribute most to the predicted probability

LIME is model-agnostic and easy to deploy, but might be unstable if the masking process is not carefully controlled. Figure 4 shows the output of LIME for a 60-second time slice. The image overall shows less constant color-fades which indicates high noise. Significant areas are still visible but more diffused. This can most likely be improved by tuning LIME parameters better to the model it is used with. In general, LIME does not work as well as other methods with a minimal implementation. Regions with high activation are disrupted by single second slices with low or even negative activation.

SHAP

SHapley Additive exPlanations (SHAP) [8] use cooperative game theory to assign each feature a Shapley value, representing its average contribution to the prediction. Deep SHAP is applied since it leverages the network's structure for efficiency to the 60×4 input, resulting in a 60×4 attribution map. Aggregating over channels or time points reveals the most influential biosignals and temporal regions. There are multiple different versions of SHAP such as KernelSHAP, TreeSHAP or GradientSHAP. DeepSHAP combines ideas from Shapley values and DeepLIFT's backpropagation rules. This makes DeepSHAP excellent at efficiency and scalability while still recovering true Shapley values exactly see [8]. Analogue to the previous Figures, Figure 5 displays Deep SHAP's activation over a 60-second slice of input data. The decision-impacting areas are marked by red stripes and are clearly visible.

Integrated Gradients

Integrated Gradients [9] attribute the model's output to its inputs by accumulating gradients along a straight line path from a baseline (for example a vector filled with zeroes) to the actual input.

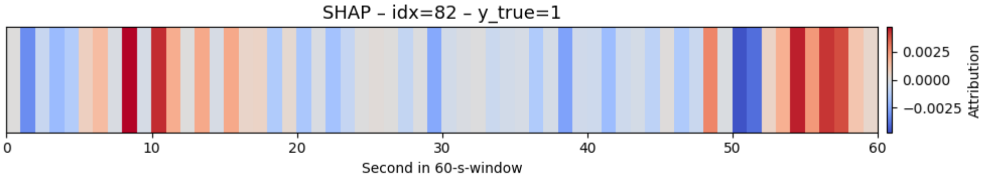


Figure 5: SHAP activation map for a single 60-second window

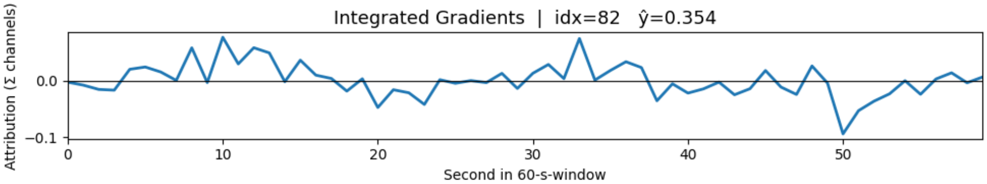


Figure 6: Integrated Gradients activation map for a single 60-second window

The corresponding integral formula is approximated with 50 discrete steps as it provides a good trade off between runtime and accuracy, producing a dense attribution map. Figure 6 shows the attribution profile of Integrated Gradients for one 60-second window. This time, the peaks are not color-coded but represented by a line-diagram.
The first plot shows the curve oscillating in both directions around zero, resulting in a predicted value of 0.354 demonstrating it spends slightly more time above zero than below zero but still predicts label 0 (no apnea).

4.1 Quantitative evaluation of explanations

To evaluate the different approaches, a benchmark of 100 held out windows (200 deletion/insertion steps, averaged over 3 repeats) to compare methods were ran on:

1. Faithfulness: Measured by Deletion AUC (lower is better) and Insertion AUC (higher is better), following [5]
2. Sparsity: The entropy of the attribution distribution (lower is sparser)
3. Latency: Average runtime per window

Insertion AUC is used to measure the quality of the explainability method. To calculate it, a baseline input, in this case a 60-second band filled with zeroes on every signal is created. Then the seconds depicted to be most influential are added sequentially. For this benchmark, only the most significant 10 % of values were added. The better the explainability model, the faster the AUC should rise. Note that this AUC is not the ROC-AUC depicted earlier but is the area under the curve of baseline frames against real frames. The Deletion AUC works similar. This time, a full, real 60-second window is chosen. The 10 % of most important seconds are removed. The AUC should this time drop as fast as possible. To measure sparsity, the Shannon entropy [10] is used. It is a numeric indicator for the average number of decisions required to identify a class from a pool of classes. In this case it is the normalized entropy of the attribution distribution. First, the maximum possible Shannon Entropy was calculated and all results were normalized, indicating how close to maximum they are. A high sparsity value means low entropy. To measure latency, the time used to calculate one full heatmap for a single window was used.
This benchmark used 100 held out subjects from SHHS1 to ensure the benchmark is not subject to irritations due to model overfitting. Table 2 shows the results of the 100-window benchmark. Grad-CAM shows low insertion AUC, low deletion AUC, fair sparsity and fast latency. This makes it the method of choice for real-time or embedded use-cases. Integrated Gradients has the best insertion AUC value, low deletion AUC, lower sparsity and is fairly fast. This makes it a good allrounder tool for applications that require both accuracy and performance. Deep SHAP has lower insertion AUC, low deletion AUC and very high sparsity but is fairly slow. It is best suited for applications that need precise peaks, for example double checking after running a rather coarse method. LIME has good insertion AUC, low deletion AUC, lowest sparsity and is very slow. This makes the method rather unsuited for most use-cases. It should only be used if a linear model or feature weights are explicitly required.
Finally, the model of choice is dependent on the individual use case and must be chosen for the project at hand. Table 3 summarizes these evaluations.

Table 2: Results of quantitative evaluation

Method	**Deletion AUC (↓)**	**Insertion AUC (↑)**	**Sparsity (↑)**	**Latency (ms, ↓)**
Grad-CAM	0.009	0.011	0.077	23
Integrated Gradients	0.008	0.166	0.049	72
Deep SHAP	0.008	0.106	0.107	225
LIME	0.008	0.157	0.030	620

Table 3: Recommended XAI method by use-case

Use Case	**Recommended Method**
Real-time / embedded	Grad-CAM
Highest faithfulness	Integrated Gradients
Maximum sparsity	Deep SHAP
Feature-weight insight	LIME

5 Conclusion

This work has demonstrated that Deep Learning models can effectively detect sleep apnea events from biosignals, achieving strong predictive performance. However, interpretability remains critical for clinical trust and adoption. By applying and comparing four XAI techniques, it showed that no single method is optimal in all respects.
Grad-CAM offers speed and reasonable clarity, making it suitable for real-time or embedded applications. Integrated Gradients is fairly fast and produces the best results regarding faithfulness while keeping medium sparsity.Deep SHAP excels in sparsity, but is computationally more intensive than Grad-CAM and Integrated Gradients while producing less faithful results. LIME and Integrated Gradients are applicable regardless of the Deep Learning model used, however come with some trade-offs in either clarity or runtime. Especially LIME should only be used if either a linear model is mandatory or the feature weights that are a by-product of computing LIME can be used beneficially. These findings highlight the need to match XAI strategy to application context. Future work should investigate hybrid explanation approaches and human-centered validation with clinicians.

Data availability statement

The dataset used in this work is publicly available here: `https://sleepdata.org/datasets/shhs` [11] [12]

References

[1] Kristiansen, Stein, Konstantinos Nikolaidis, Thomas Plagemann, Vera Goebel, Gunn Marit Traaen, Britt Øverland, Lars Aakerøy, Tove-Elizabeth Hunt, Jan Loennechen, Sigurd Steinshamn, Christina Bendz, Ole-Gunnar Anfinsen, Lars Gullestad, and Harriet Akre. (2021) "Machine Learning for Sleep Apnea Detection with Unattended Sleep Monitoring at Home", *ACM Transactions on Computing for Healthcare* **2**: 1–25. doi:10.1145/3433987.

[2] Á. Serrano Alarcón, N. Martínez Madrid, R. Seepold and J. A. Ortega, "Obstructive sleep apnea event detection using explainable Deep Learning models for a portable monitor", *Frontiers in Neuroscience*, Bd. 17, Art. 1155900, 2023. doi:10.3389/fnins.2023.1155900

[3] A. John, B. Cardiff and D. John, "A 1D-CNN Based Deep Learning Technique for Sleep Apnea Detection in IoT Sensors", CoRR, abs/2105.00528, 2021. arXiv:2105.00528

[4] F. Doshi-Velez and B. Kim, "Towards A Rigorous Science of Interpretable Machine Learning", arXiv preprint, abs/1702.08608, 2017. arXiv:1702.08608, doi:10.48550/arXiv.1702.08608

[5] V. Petsiuk, A. Das and K. Saenko, "RISE: Randomized Input Sampling for Explanation of Black-box Models", CoRR, abs/1806.07421, 2018. arXiv:1806.07421

[6] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh and D. Batra, "Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization", CoRR, abs/1610.02391, 2016. arXiv:1610.02391. doi:10.48550/arXiv.1610.02391

[7] M. T. Ribeiro, S. Singh and C. Guestrin, "Why Should I Trust You?: Explaining the Predictions of Any Classifier", in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, San Francisco, CA, USA, 2016, S. 1135–1144. doi:10.1145/2939672.2939778

[8] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions", in *Advances in Neural Information Processing Systems 30 (NIPS '17)*, 2017, S. 4768–4777. doi:10.5555/3295222.3295230

[9] M. Sundararajan, A. Taly and Q. Yan, "Axiomatic Attribution for Deep Networks", in *Proceedings of the 34th International Conference on Machine Learning (ICML '17)*, 2017, S. 3319–3328. doi:10.5555/3305381.3305518

[10] C. E. Shannon, "A Mathematical Theory of Communication", *Bell System Technical Journal*, Bd. 27, Nr. 3, S. 379–423, 1948. doi:10.1002/j.1538-7305.1948.tb01338.x

[11] S. F. Quan et al., "The Sleep Heart Health Study: design, rationale, and methods", *Sleep*, Bd. 20, Nr. 12, S. 1077–1085, Dez. 1997. PMID: 9493915

[12] G.-Q. Zhang, L. Cui, R. Müller, S. Tao, M. Kim, M. Rueschman, S. Mariani, D. Mobley and S. Redline, "The National Sleep Research Resource: towards a sleep data commons", *Journal of the American Medical Informatics Association*, Bd. 25, Nr. 10, S. 1351–1358, 2018. doi:10.1093/jamia/ocy064; PMID: 29860441; PMCID: PMC6188513