Audio sleep stage binary classification based on Traditional and Deep features. Comparison.

1. Intro

To effectively classify sleep, numerous signals and their characteristics can be analyzed. One way to do it is by recording sounds. Various basic research revealed that the dynamics of respiration [1], [2] relate to sleep stages. Nocturnal sounds, which can be easily recorded throughout the night, contain rich information about sleep. These sounds provide insights into respiratory patterns, sleep-activity patterns, and variable breathing sounds corresponding to changes in muscle tone in the upper airway. Developing such model could help individuals understand and measure the overall level of their sleep quality, and detect sleep disruptions.

This project explores binary sleep stage classification—awake versus asleep—based on two primary feature extraction approaches: traditional manual extraction and deep learning with Convolutional Neural Networks (CNNs). By comparing these methods, the aim is to evaluate their effectiveness and potential applications in non-invasive sleep monitoring solutions.

2. Dataset

PSG-Audio [3] – The dataset contains hospital ambient sounds of 212 sleep apnea patients recorded with a microphone placed approximately 1m above the patient's bed. Sounds are sampled at 48 KHz as 24-bit .wav files. Additional non-sleeping sounds are collected, such as various domestic sounds: clock ticking, silence/background noise recordings, street sound ambience, quiet rain, thunderstorm sounds, people talking and background television sounds.
The dataset is manually labelled based on respiratory patterns [Figure 2]. Given the dataset's bias toward sleep apnea patients, preprocessing steps were undertaken to ensure it could generalize to a broader population.

3. Preprocessing and annotation

The first preprocessing step is loading the sound file and dividing it into 30 second fragments. A 200Hz High Pass filter is applied to reduce some of the hum. A spectral gate noise reduction algorithm is applied to minimize background noise. The next step is turning each fragment into a mel spectrogram. A spectrogram is a representation of the spectrum of frequencies of a signal as it varies with time. A mel spectrogram is a spectrogram where frequencies are converted to the mel scale, which better fits human hearing. They also emphasize the breathing pattern which is more stable and independent from overall sound amplitude. The frequency domain of each mel spectrogram is set at 20 frequency bins, while the time steps are 1407. The mel spectrograms are randomly pitch shifted in the range (-0.2, 0.2) for data augmentation, simulating different types of respiratory events.

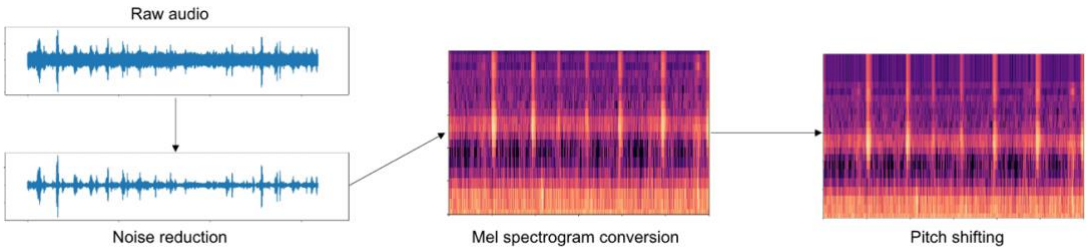
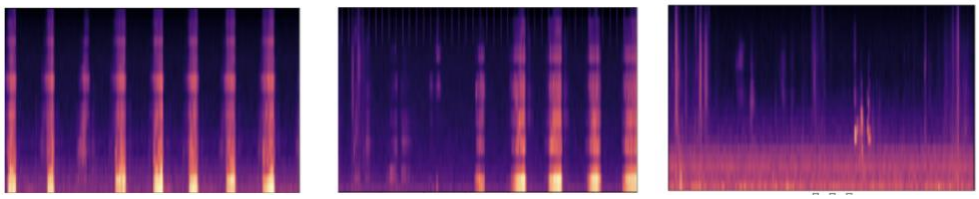


Figure 1. *Preprocessing procedure [4]*

Annotation criteria: For a supervised learning approach, we need sleep classifications. As we don't have official sleep labels in the dataset the solution is manual annotation. The labels will be 0 (Awake) and 1 (Asleep).

For a sleep activity to occur the following criteria is needed:

a) A constant periodic breathing pattern
b) The breathing pattern must last for more than 50% of the fragment (at the start and/or at the end of the fragment
c) If the breathing is nonexistent or it lasts for less than 50% of the fragment, it's classified as 0 (Awake)



a) *b)* *c)*

Figure 2. *Normal sleep activity*

What's known is that people with sleep apnea pause for over 10 seconds while they breathe, take shallow breaths, gasp, or choke. Using that information, fragments of unhealthy breathing are easily found [Figure 3.] and most of them are discarded to avoid training bias.

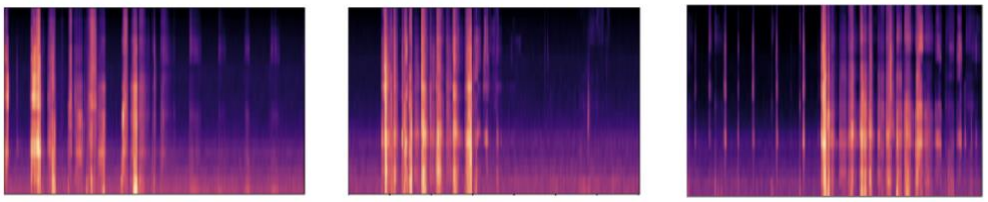


Figure 3. *Sleep apnea fragments. That's expressed when the fragment starts with silence and the breathing pattern is in the middle, followed by silence. Some irregularity in the breathing might be found.*

We are left with a total of 55000 fragments of mel spectrograms along with their classifications. To save space, the fragments are saved as NumPy array files (.npy). 31140 of them were classified as awake and 23860 as asleep.

4. Methods

To classify the sleep stages, two approaches were implemented: a traditional manual feature extraction method with a Support Vector Machine (SVM) classifier and a deep learning method using a Convolutional Neural Network (CNN).

Manual feature extraction

The fragment is converted to an envelope curve. A peak detection algorithm is applied where the threshold for peak detection is set dynamically as the mean plus one standard

deviation of the envelope. The autocorrelation of the filtered envelope is computed to analyze periodicity. (Figure 4.) Autocorrelation is essentially the correlation between the signal and its time-lagged version.

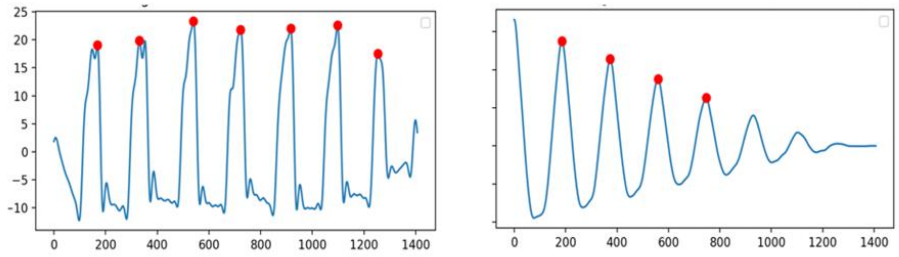


Figure 4. *Raw envelope -> Autocorrelation. The red dots are the peaks.*

A total of 10 features were extracted. These features include statistics of detected peaks, time intervals between peaks, and autocorrelation analysis. [5], [6]
The initial extracted features were 14 but 4 were removed due to a significant correlation between them caused by their similar calculation methods. (Figure 5.)

Feature	**Description**
Var coefficient	Coefficient of variation of envelope. Standard deviation divided by the mean.
Skewness	The asymmetry of the distribution of the amplitude envelope
Crossings	Number of threshold crosses
Breath peaks	Number of signal peaks
Sum of peaks	Sum of all peak amplitudes
Range of peaks	Difference between the minimum and maximum peak
Avg interval	Average time interval between peaks [5]
Stdev interval	Standard deviation of time intervals [5]
Cycle period	Average respiratory cycle duration (Autocorrelation first peak timestep) [6]
Cycle intensity	First peak amplitude of autocorrelation. A measure of strength of the periodic pattern. [6]

Table 1. *Final selected features for classification*

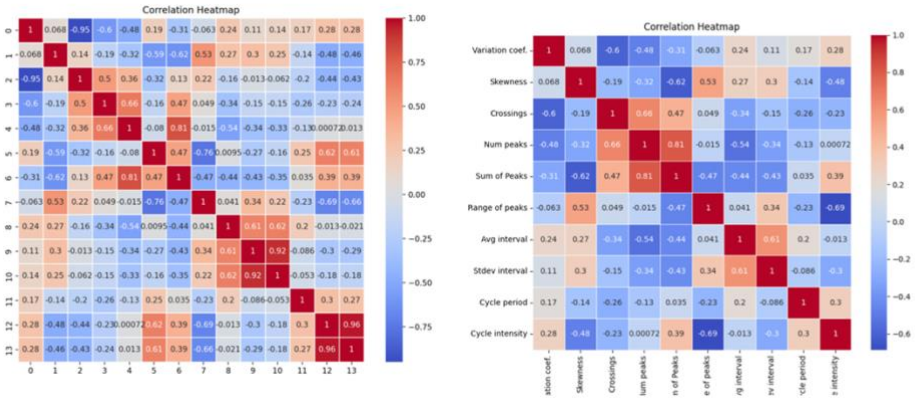


Figure 5. *Correlation Heatmap of extracted features before and after filtering. The same accuracy is achieved with both feature sets.*

A Z-Score normalization is applied after extraction to minimize outlier influence on training. In Z-Score the formula *"(x – μ) / σ"* is used, where *x* is the original value, *μ* is the mean of feature data and *σ* is the standard deviation of feature data.
The manual features are classified using a Support Vector Machine classifier. SVMs can handle high-dimensional data (10 features) and are particularly good at solving binary classification problems. Here, the scikit-learn SVM package is employed to train a binary classifier with radial basis function (RBF) kernel to map the features into non-linear space.

Convolutional Neural Network extraction
For CNN extraction, Google's Xception [7] architecture is used due to its high prediction rate for similar tasks [8]. The overall architecture of Xception network contains three main flows, i.e., entry, middle, and exit flows. The middle flow is sometimes referred to as the core structure part, and it comprises a 9-layer structure that repeats 8 times. Within the 9-layer structure, there are 3 layers each of Relu, separable Conv2D, and batch normalization. To improve the architecture performance, the core structure is changed to repeat 3 times as shown in *A. Mehmood's* work [9]. That would significantly simplify the model without affecting accuracy. (Figure 6.)

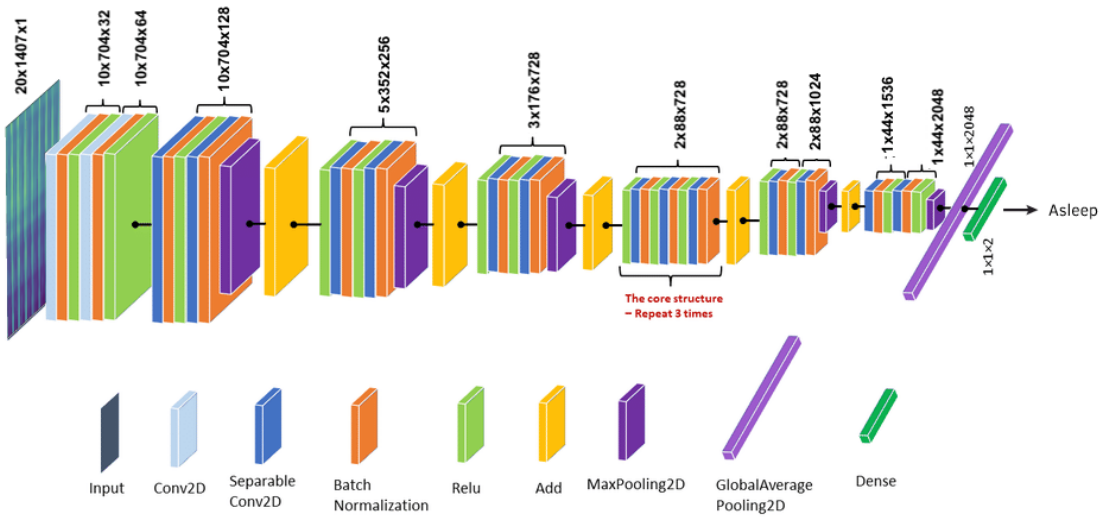


Figure 6. *Proposed structure of Xception network used within each stream of CNN. [7] [9]*

The model inputs a 20x1407 mel spectrogram fragment and outputs 2048 high level features that are fed through a fully connected layer with a Sigmoid activation function. (Figure 7.)

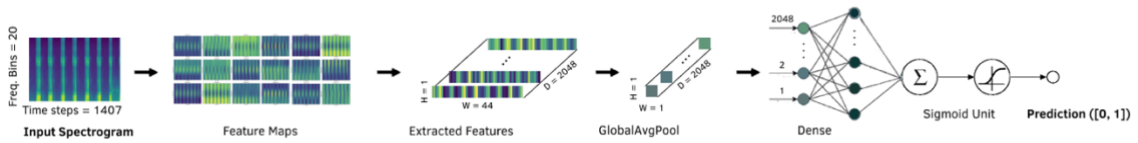


Figure 7. *Visualization of the feature extraction and classification process*

The model is trained using the Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 0.01. Performance is monitored using cross-entropy loss and macro F1-score. The train/validation/test split is 80/10/10.

5. Results

Before analysing the results, let's visualize an example of a whole night's sleep. A hypnogram plot was generated using a 7.2 hour personal audio recording, consisting of 867 sequentially predicted fragments. The plot illustrates the algorithm's confidence levels for each prediction, where a value of 0 indicates 100% confidence that the fragment corresponds to 'Awake' sounds, and a value of 1 indicates 100% confidence that the fragment corresponds to 'Asleep' sounds. Gaussian filtering was applied to smoothen the data.

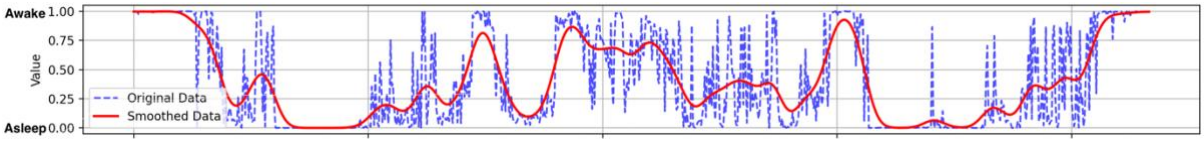


Figure 8. *My sleep recording, predicted with the Xception CNN model. The graph essentially represents periodic breathing activity during the night.*

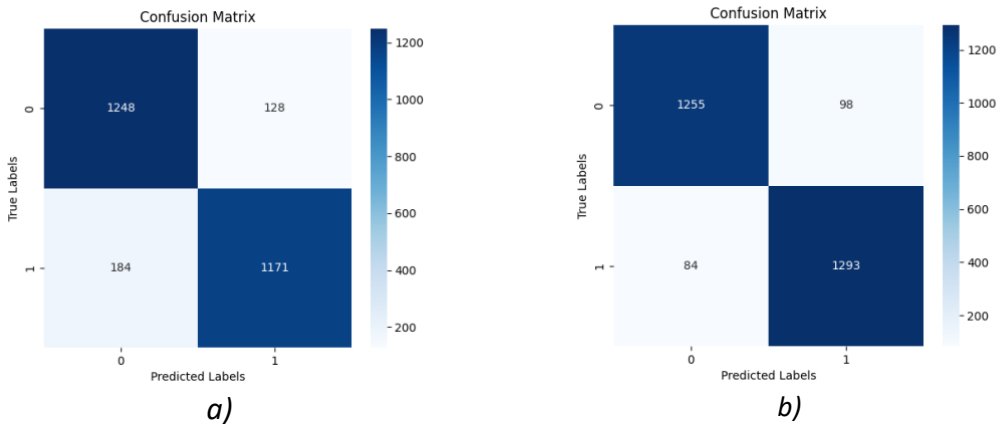
Various sleep quality indicators could be calculated using this data, such as:

Sleep Efficiency	Ratio of time spent asleep (above 0.5) to total time in bed.
Sleep Onset Latency	Time it takes to fall asleep after getting into bed.
Total Sleep Time (TST)	Total duration spent asleep (above 0.5)
Wake After Sleep Onset (WASO)	The total number of minutes that a person is awake after having initially fallen asleep

Table 2. *Potential sleep quality indicators that could be calculated with models' output*

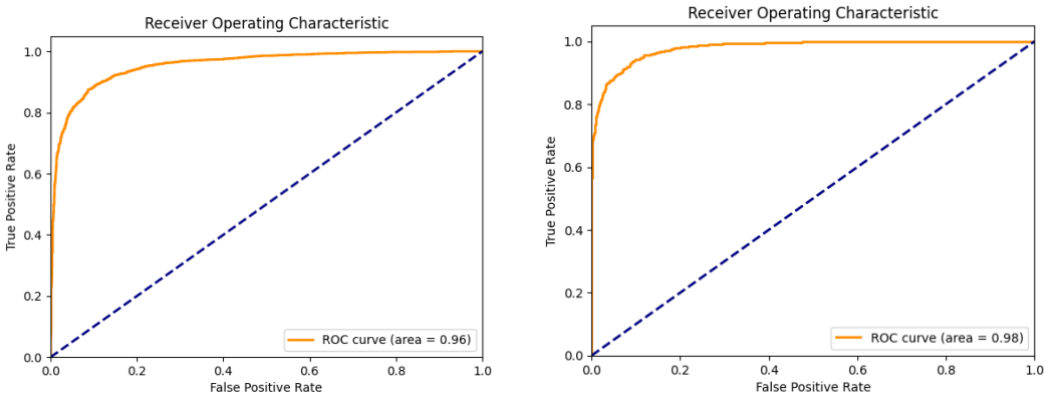
Performance evaluation:

To evaluate the effectiveness of each classification approach, several performance metrics were analysed: confusion matrices, ROC AUC, accuracy and prediction time.



a) *b)*

Figure 9. *Confusion Matrix of a) Manual Classifier; b) CNN Classifier*



a) *b)*

Figure 10. *Receiver Operating Characteristic of a) Manual Classifier; b) CNN Classifier*

	Accuracy (%)	**Prediction time (ms)**
SVM (rbf kernel)	89	0.43
CNN (Xception)	93	61

Table 3. *Performance measures of both algorithms*

The SVM model demonstrates good performance with high accuracy and high ROC AUC (0.96). Its fast prediction time suggests it is well-suited for real-time or resource-limited applications. However, it shows slightly higher misclassification rates than the Xception approach.
The Xception model achieves even higher accuracy than the SVM, along with an improved ROC AUC (0.98), indicating robust classification ability and lower misclassification rates. However, with a prediction time of 61 ms per fragment, this model may be more suitable for batch processing rather than real-time applications due to the higher computational demand.

Assuming that a periodic breathing pattern correlates to sleep activity, recording nocturnal sounds and using classification models is a valid approach to measure an individual's sleep quality. This comparison highlights the trade-off between accuracy and computational efficiency. Even though Xception model provides superior classification performance, the SVM model remains the most practical option for applications where rapid inference is critical, offering faster prediction time while classification accuracy is still reasonably high.

[1] Douglas NJ, White DP, Pickett CK, Weil JV, Zwillich CW. Respiration during sleep in normal man. Thorax. 1982 Nov;37(11):840-4. DOI: 10.1136/thx.37.11.840

[2] Snyder F, Hobson JA, Morrison DF, Goldfrank F. Changes in respiration, heart rate, and systolic blood pressure in human sleep. J Appl Physiol. 1964 May;19:417-22. DOI: 10.1152/jappl.1964.19.3.417

[3] Korompili, G., Amfilochiou, A., Kokkalas, L. *et al.* PSG-Audio, a scored polysomnography dataset with simultaneous audio recordings for sleep apnea studies. *Sci Data* **8**, 197 (2021). https://doi.org/10.1038/s41597-021-00977-w

[4] Hong J, Tran HH, Jung J, Jang H, Lee D, Yoon IY, Hong JK, Kim JW. End-to-End Sleep Staging Using Nocturnal Sounds from Microphone Chips for Mobile Devices. *Nat Sci Sleep*. 2022;14:1187-1201 https://doi.org/10.2147/NSS.S361270

[5] Kalkbrenner C, Brucher R, Kesztyüs T, Eichenlaub M, Rottbauer W, Scharnbeck D. Automated sleep stage classification based on tracheal body sound and actigraphy. Ger Med Sci. 2019 Feb 22;17:Doc02. doi: 10.3205/000268. PMID: 30996721; PMCID: PMC6449867.

[6] Dafna E, Tarasiuk A, Zigel Y. Sleep-wake evaluation from whole-night non-contact audio recordings of breathing sounds. PLoS One. 2015 Feb 24;10(2):e0117382. doi: 10.1371/journal.pone.0117382. PMID: 25710495; PMCID: PMC4339734.

[7] Deep learning with depthwise separable convolutions,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1800–1807.

[8] Wasin Kalintha, Takafumi Kato, Ken–ichi Fukui, SleepAge: Sleep Quality Assessment from Nocturnal Sounds in Home Environment, Procedia Computer Science, Volume 176, 2020, Pages 898-907, ISSN 1877-0509, doi: 10.1016/j.procs.2020.09.085.

[9] A. Mehmood, "Efficient Anomaly Detection in Crowd Videos Using Pre-Trained 2D Convolutional Neural Networks," in *IEEE Access*, vol. 9, pp. 138283-138295, 2021, doi: 10.1109/ACCESS.2021.3118009