Sleep Dtages Prediction

Extracting Features VS Using Deep Models

on Monodimensional (multi-channel) Signal

Digital Signal and Image Management Project
University of Milano-Bicocca

Matteo Breganni 869549
Francesco Cavallini 920835

UNIVERSITA' DEGLI STUDI DI MILANO BICOCCA

Brief Introduction on Dataset

Dataset Origin

There is a total of **44 ST* signals** (ST = Sleep Telemetry). Those were obtained in a 1994 study of temazepam effects. Those 44 files were registered on:

- **22 subjects** (males and females)
- Over the range of **2 consecutive nights**

Each of these 44 recordings is 9h long and were then **split into 2 files** for a total of:

- **44 "...-PSG.edf"** files as our EEG multi-channel signal:
  - 4 Channels: EEG Fpz-Cz, EEG Pz-Oz, EOG horizontal, EMG submental. All sampled at 100 Hz
  - +1 Channel: event markers were sampled 1 Hz

- **44 "...-Hypnogram.edf"** files as our labels to find sleep stages:
  - Stage Wake: when subject is still awake
  - Stage N1 (Light Sleep) : Transition from wakefulness to sleep.
  - Stage N2 (Stable Sleep): This stage is the largest portion of sleep, about 50% of the night.
  - Stage N3 & N4 (Deep Sleep, Stages): Characterized by slow-wave activity.
  - Stage REM (Dream Sleep): Associated with dreaming, muscle paralysis, and rapid eye movements. It features low-voltage mixed-frequency brain waves and fluctuating heart/respiration rates.

Cutting Useless Channels

After a quick research on the web those are the informations that arise about each of the 5 channels of our EEG:

1. **EEG Fpz-Cz**: This is a standard EEG channel that records brain activity. → **we keep it** as an EEG channel as it could be important to learn on

2. **EEG Pz-Oz**: Another standard EEG channel. → **we keep it** as an EEG channel as it could be important to learn on

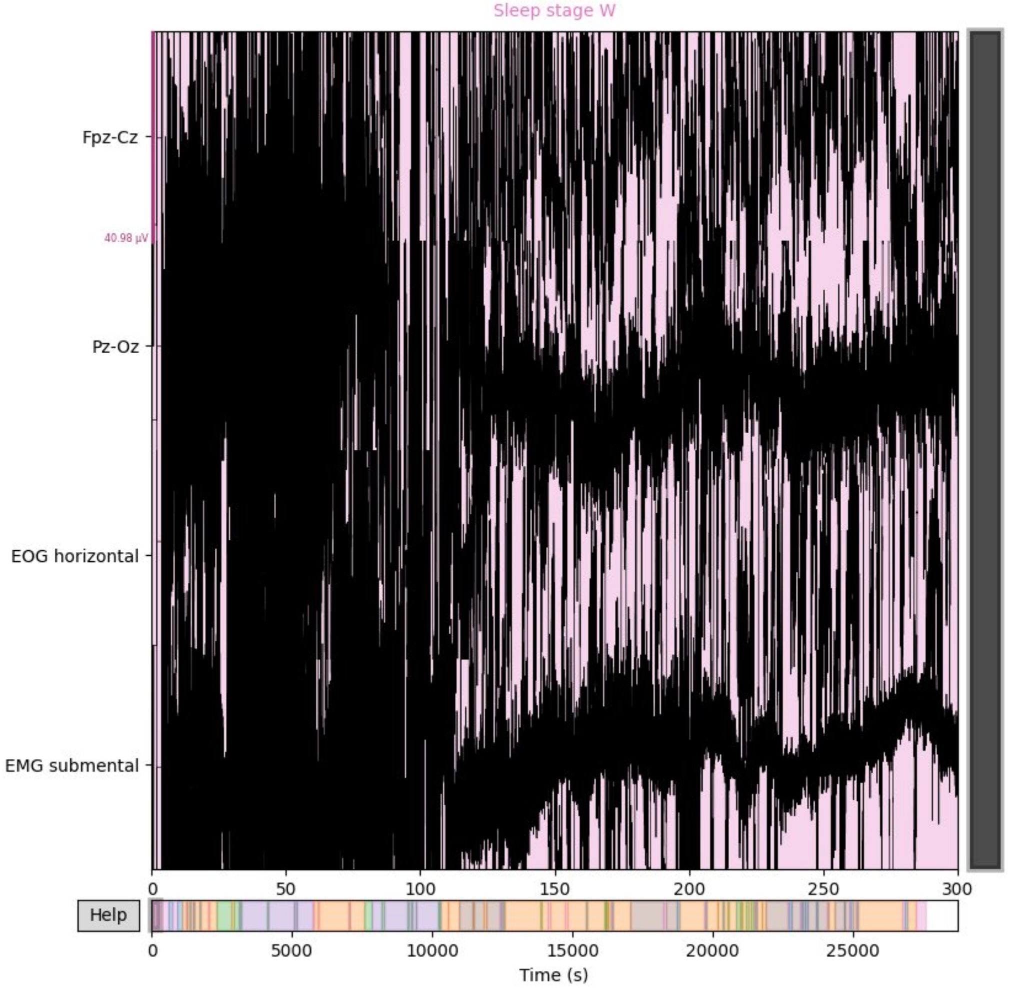
3. **EOG horizontal**: Electrooculography (EOG) is used to monitor eye movements. Eye movements are crucial for detecting REM sleep → **we keep it** as an EEG channel as it is very clearly much important to learn on (in our use-case)

4. **EMG submental**: Electromyography (EMG) from submental (chin) muscles. Submental EMG is vital for detecting muscle tone, which helps differentiate between REM sleep (low muscle tone) and other stages → **we keep it** as this channel is useful too to analyze sleep stages

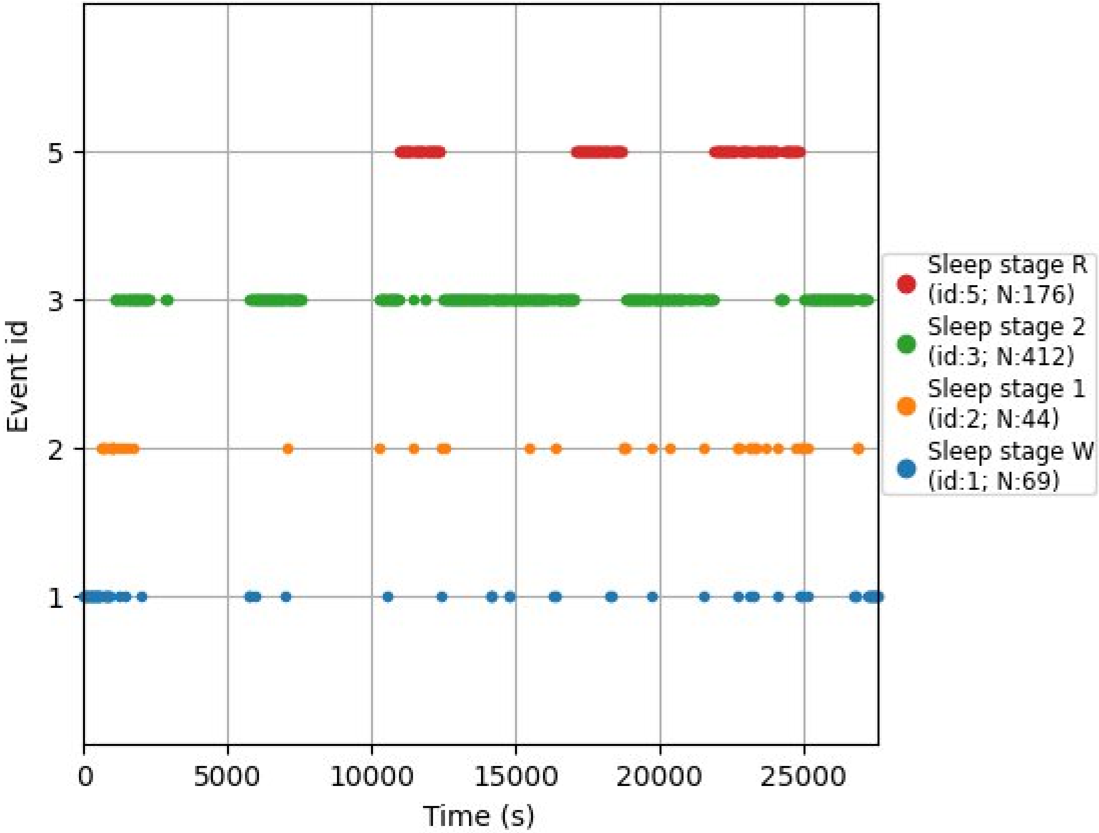
5. **Marker**: Typically used for event markers or annotations. → **we can scrap this** channel

Clean EEG Singal example

Signals:



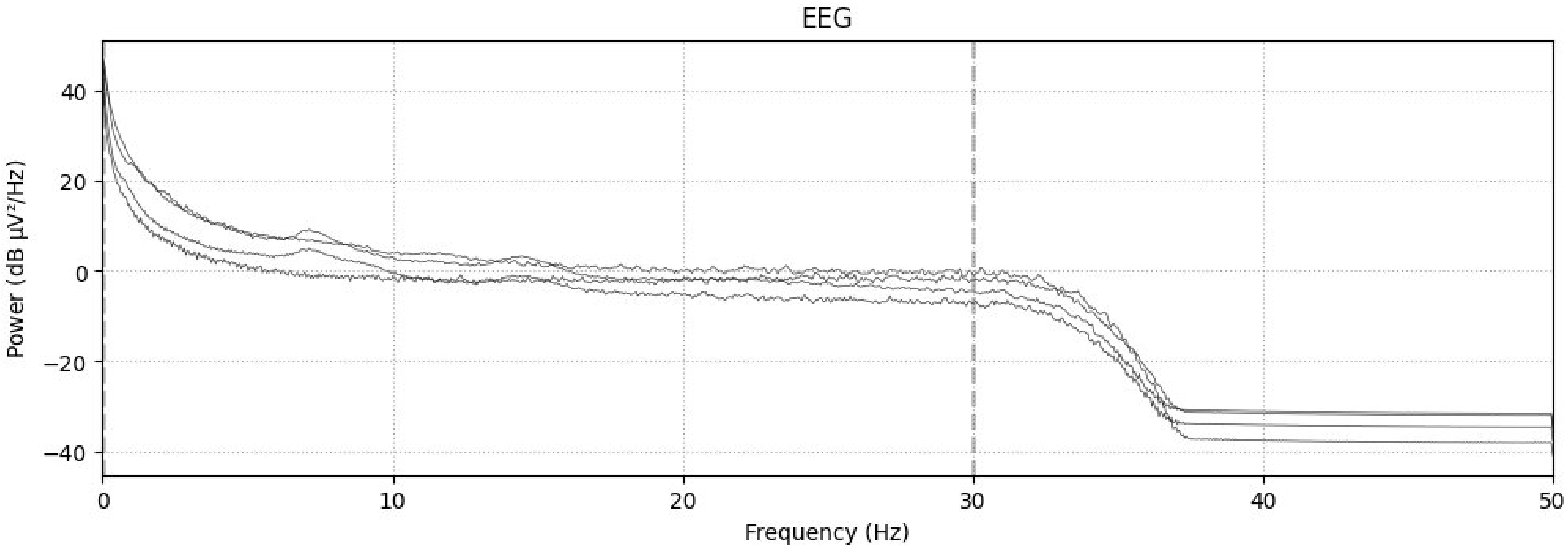
Target events:



Data Preparation

Filtering Signal

Following a brief online search, it was determined that the most relevant information in sleep EEG data resides **below 30 Hz**. To reduce the influence of higher-frequency noise, we apply a low-pass filter **with a cutoff frequency of 30 Hz** to our recordings.

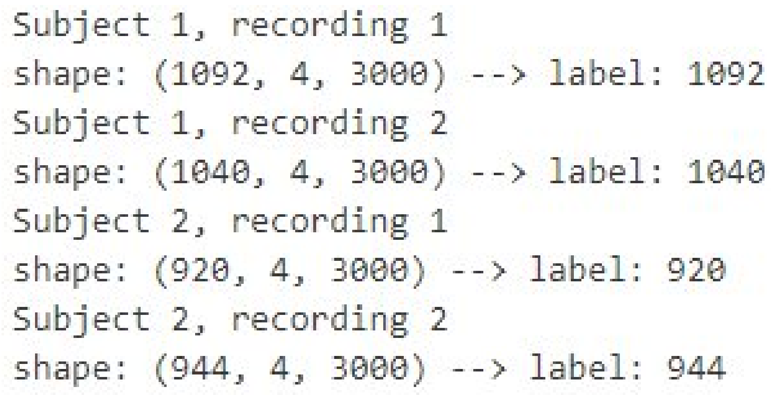


Extract Epochs

EEG epoching is a procedure in which specific time-windows (in our case 30s) are extracted from the continuous EEG signal. These time windows are called "epochs".

Extracting epochs from EEG data that means transfornming the data:

- **from in a matrix [channel x time]** fromat (where time is the complete continuous EEG signal = 8h),
- **to a matrix [epochs x channel x time]** format (where time is the time length of each epoch=30s, and epochs is the number of segments we extracted from continuous EEG signal)

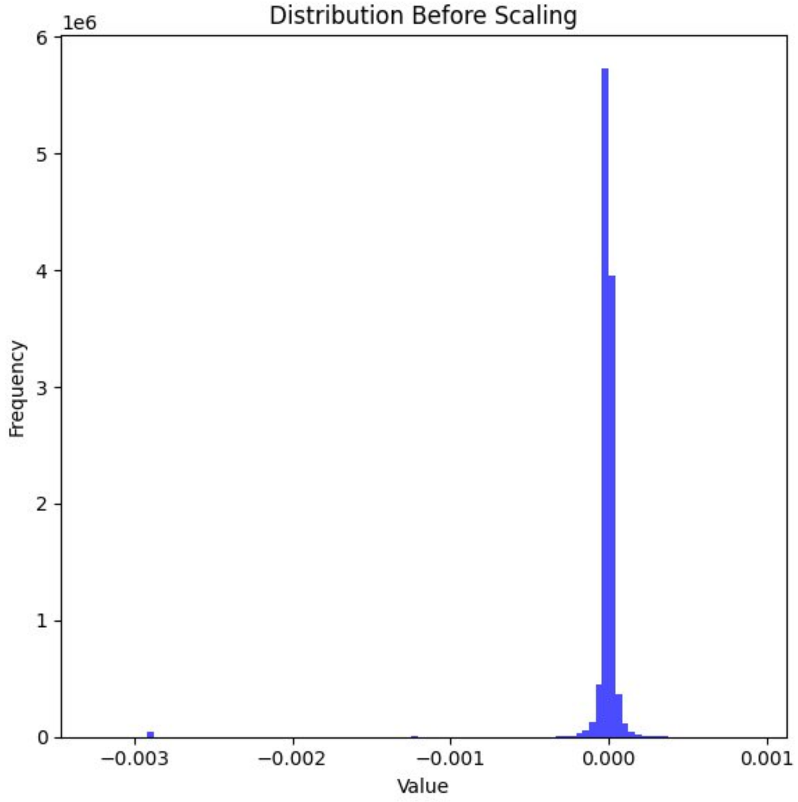


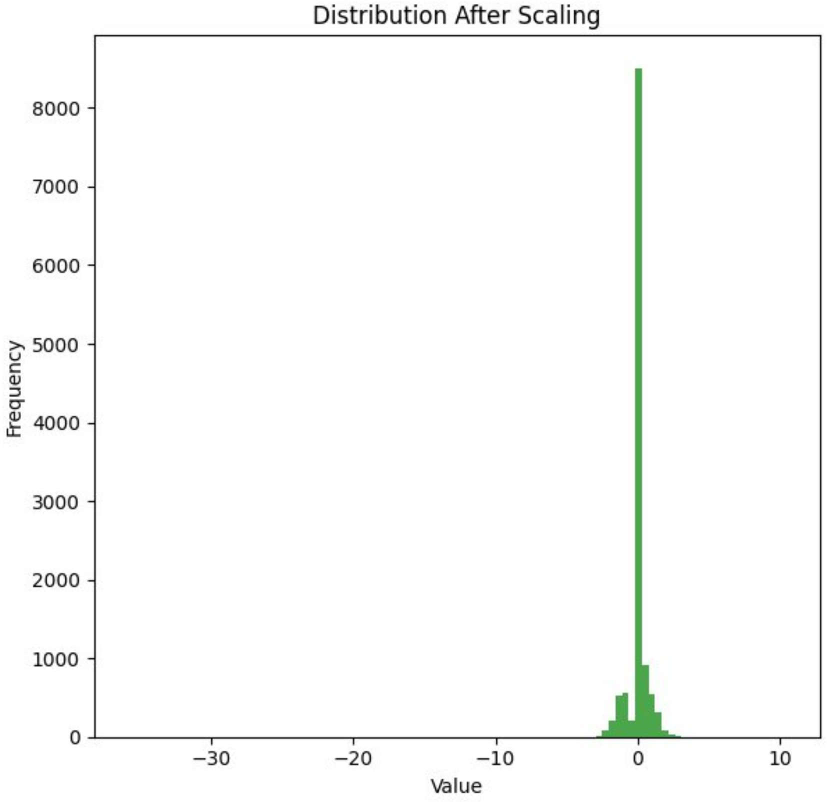
```
Subject 1, recording 1
shape: (1092, 4, 3000) --> label: 1092
Subject 1, recording 2
shape: (1040, 4, 3000) --> label: 1040
Subject 2, recording 1
shape: (920, 4, 3000) --> label: 920
Subject 2, recording 2
shape: (944, 4, 3000) --> label: 944
```

This should help generalization capabilities of all models

Standard Scaling

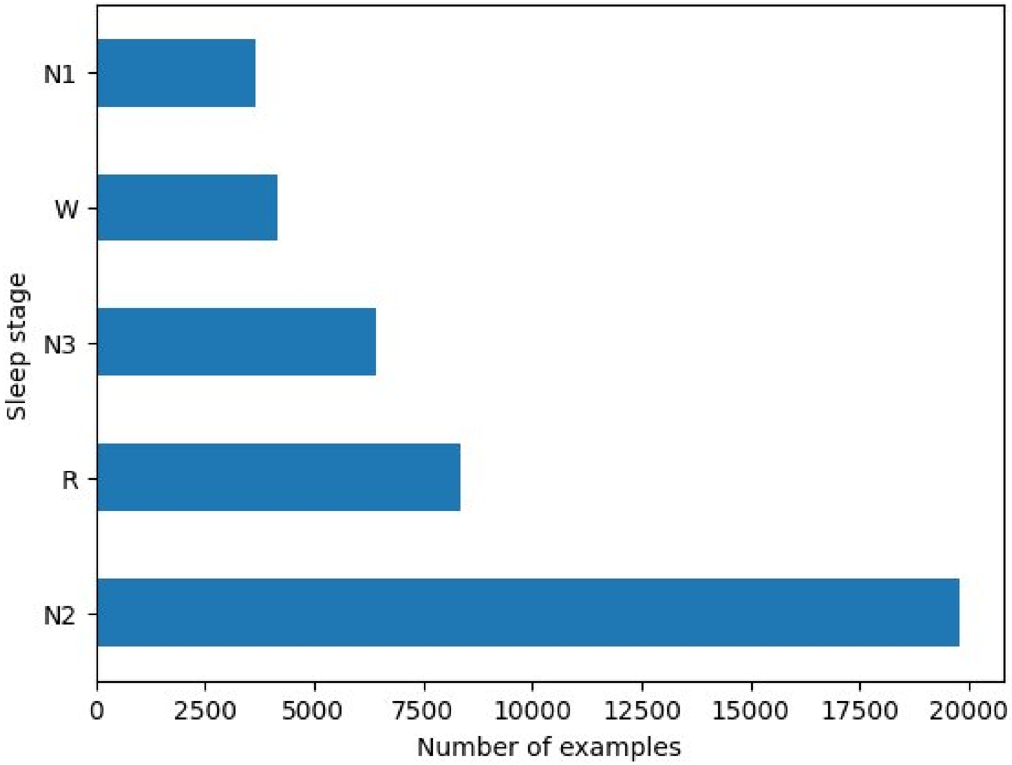
Scaling is often beneficial when trying to learn data on deep models, for this reason we apply scaling on each epoch:





Weights Balancing

Since our goal is to classify and predict using features extracted from EEG epochs, we take a look at number of epochs per label, to find out they are heavily unbalanced. for this reason we compile some weight to associate with each future training instance:



– Wake class weight:	2.03957631
– N1 class weight:	2.31929921
– N2 class weight:	0.42809358
– N3/N4 class weight:	1.32071707
– REM class weight:	1.01478021

Features extractions
&
Learning models

Models Overview

After preparing the data and splitting the full dataset into train/test/validation subsets (each containing epochs and labels) we are finally ready to start feature extractions and comparisons between different models. The methods proposed at this stage are as follows:

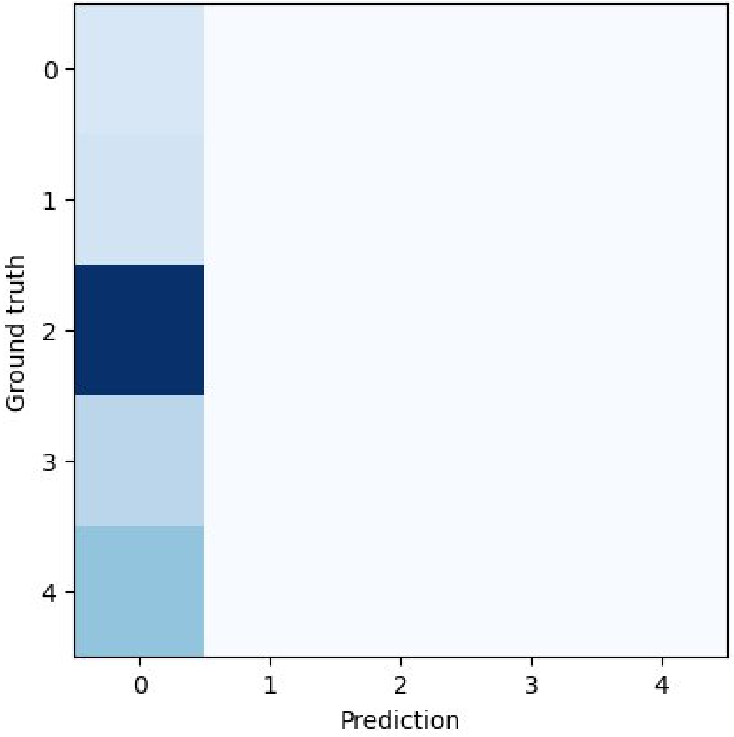
1. Extracting time domain features and learning via SVM model
2. Extracting frequency domain features and learning via SVM model
3. Combining Time Domain and frequency domain features and learning via SVM model
4. Processing identity features (raw data) via RNN
5. Processing identity features (raw data) via Chambon et al CNN

In the next sections we will explore each of these points in more detail.

Model-1: SVM + Time Domain Features

For this model we extract some time-domain features for each epoch and train an SVM model on them. Those are the extracted feature for each epoch:

- **mean value**
- **variance**
- **stardard deviatiation**
- **energy (rms)**



Hypothetically these basic statistical features **help capture the overall activity levels and variability in the EEG signal**, those could vary across different sleep stages; yet **we do not expect to get great performance from this attempt** as the feature we are extracting are very general and **have very little to none correlation to the target**. Considering also we are using a very simple model like SVM we expect bad classfication performances.

Our predictions are later confirmed correct as model doesn't seem capable to classify any class, it simply categorized everything to class 0 (wake). For this reason **this model is scrapped**

Model-2: SVM + Frequency Domain Features

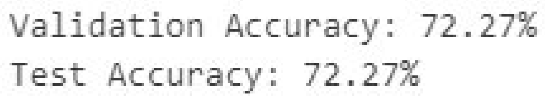
For attempt-2 we made a new SVM model and trained it on extracted frequency domain features. After a quick web research those kind of features were selected as highly correlated to EEG sleep classification targets:

- **Power Spectral Density (PSD)**: This is because different sleep stages are associated with distinct frequency bands. For example:
  - Delta (0.5-4 Hz): Deep sleep (correlates to stages N3 / N4).
  - Theta (4-8 Hz): Light sleep (correlates to stage N1).
  - Alpha (8-13 Hz): Relaxation (correlates to stage N2)
  - Beta (13-30 Hz): Wakefulness or REM (correlates to stages ”wake” / “REM”)

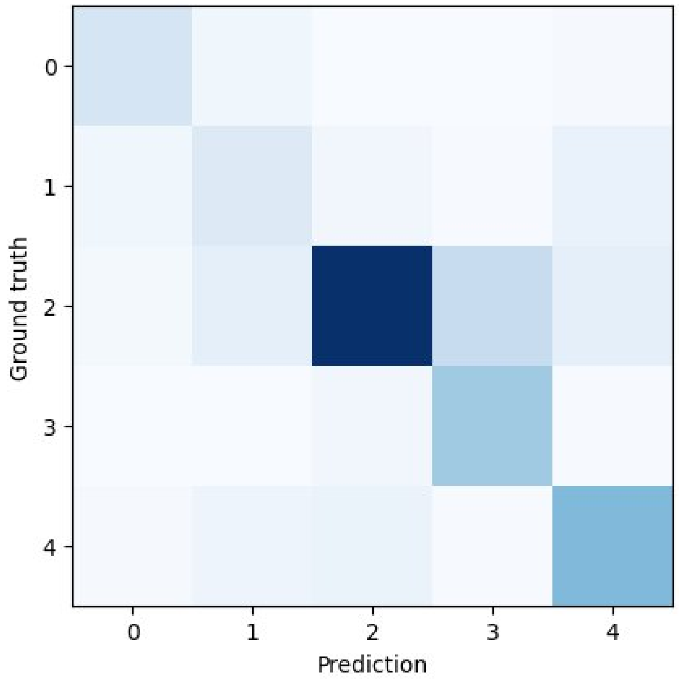
- **Spectral Entropy**: Indicates the regularity and complexity of the signal in the frequency domain, usually has lower values during deep sleep stages.

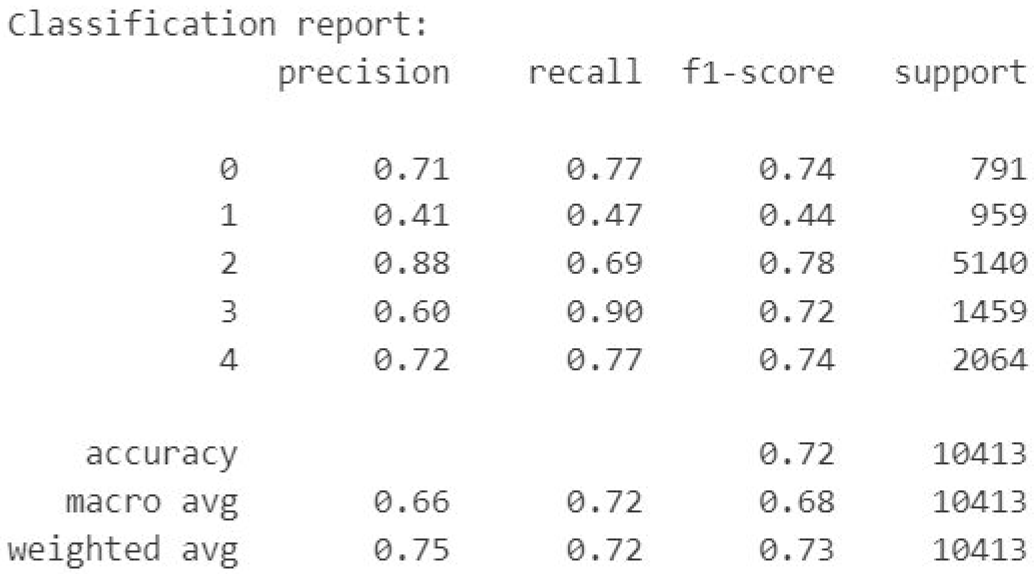
As those features are theoretically highly correlated to target, we expect this model to gain good accuracy over previous attempt.

Model-2: Results



```
Validation Accuracy: 72.27%
Test Accuracy: 72.27%
```





```
Classification report:
              precision    recall  f1-score   support

           0       0.71      0.77      0.74       791
           1       0.41      0.47      0.44       959
           2       0.88      0.69      0.78      5140
           3       0.60      0.90      0.72      1459
           4       0.72      0.77      0.74      2064

    accuracy                           0.72     10413
   macro avg       0.66      0.72      0.68     10413
weighted avg       0.75      0.72      0.73     10413
```

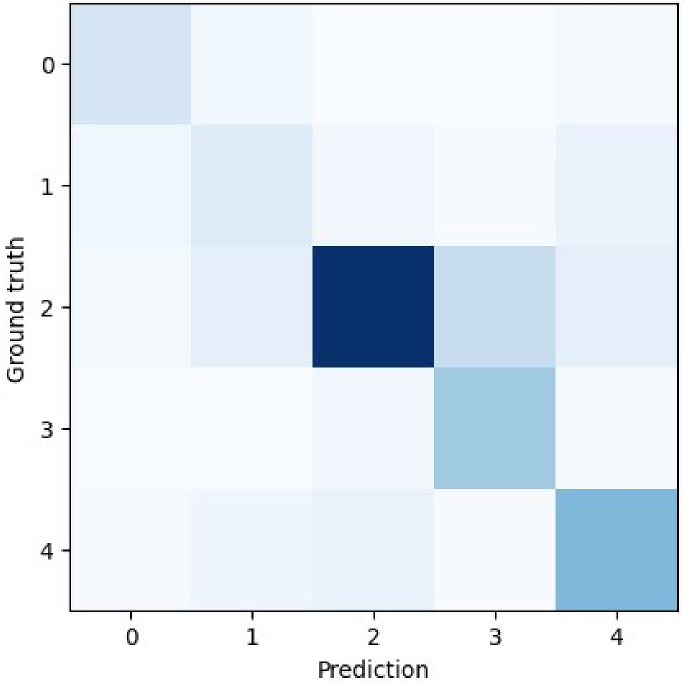
We can now see a HUGE jump in performance, we are able to classify this hard problem whit just a very simple model and having more than 70% certainty in both accuracy and f1-score.

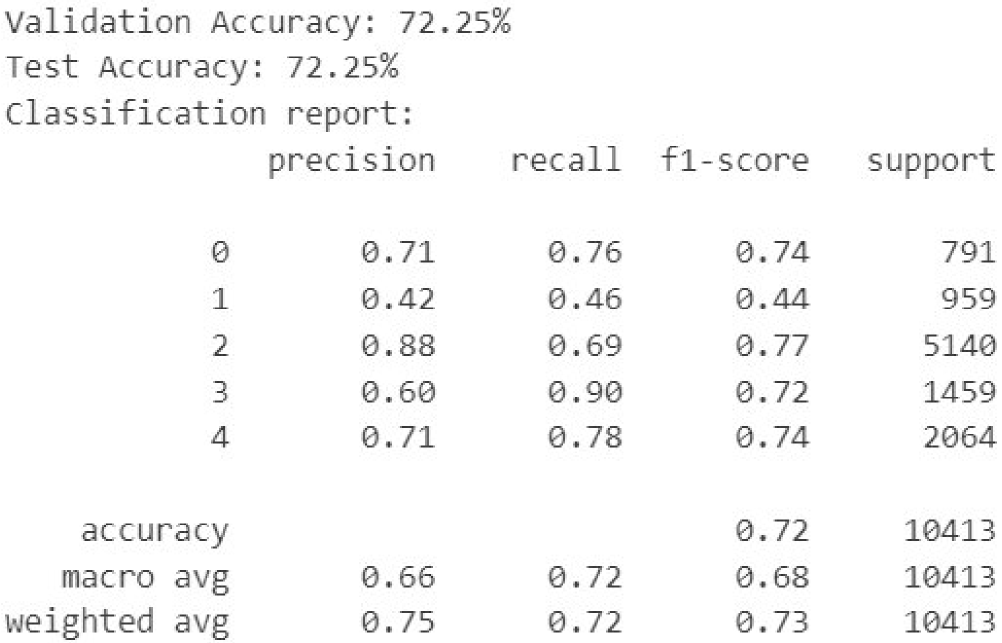
From confusion matrix alone we can clearly see that the one class we are able to predict most confidently is the class 2, wich makes perfect sense as it is the one whit most samples and it also makes sense that class 2 and 3 are the following most precise as Spectral Entropy might help classify those the most.

This results alone prove this problem can be solved even by a simple model like SVM;

Model-3: Time + frequency features combined

We achieved high accuracy with frequency domain features. Now, we **aim to enhance accuracy by incorporating features from model-1**, like adding missing puzzle pieces for a clearer picture. Results are as follows:





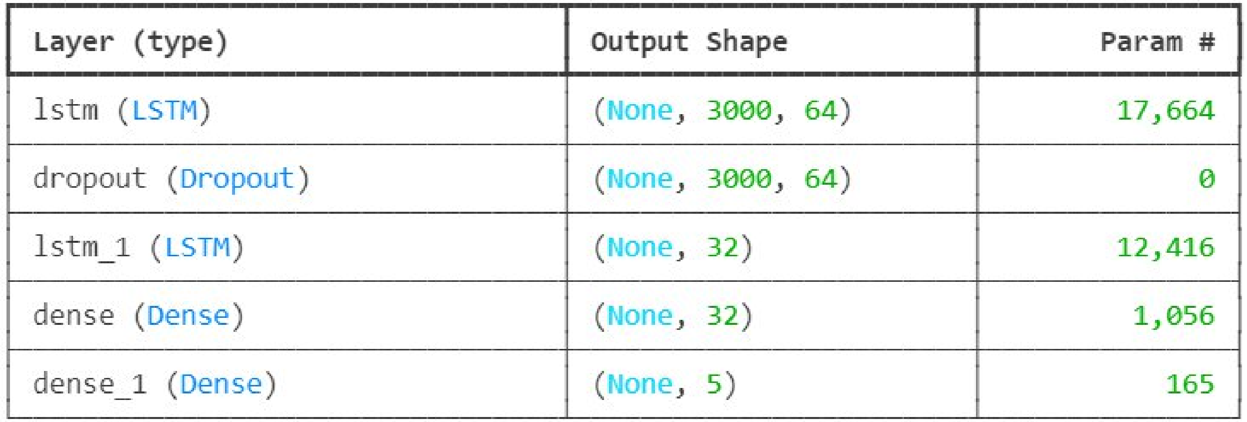
```
Validation Accuracy: 72.25%
Test Accuracy: 72.25%
Classification report:
              precision    recall  f1-score   support

           0       0.71      0.76      0.74       791
           1       0.42      0.46      0.44       959
           2       0.88      0.69      0.77      5140
           3       0.60      0.90      0.72      1459
           4       0.71      0.78      0.74      2064

    accuracy                           0.72     10413
   macro avg       0.66      0.72      0.68     10413
weighted avg       0.75      0.72      0.73     10413
```

The accuracy very so slightly decreased by 0.02%, suggesting two possibilities:

- The SVM likely ignored the added time-domain features (most likely option)
- The additional features may have improved generalization. (unlikely option as **0.02% is too little of a change to see any actual difference**)

Model-4: RNN on Raw Data

Recursive Neural Networks (RNN) are **designed to process sequential data.** This means RNNs could be very useful in our use-case as all of our sleep stages samples are non-other that sequential data. Also we could benefit knowing previous sleep stages to classify current one; which means that, on paper, RNN should be very good on this task.

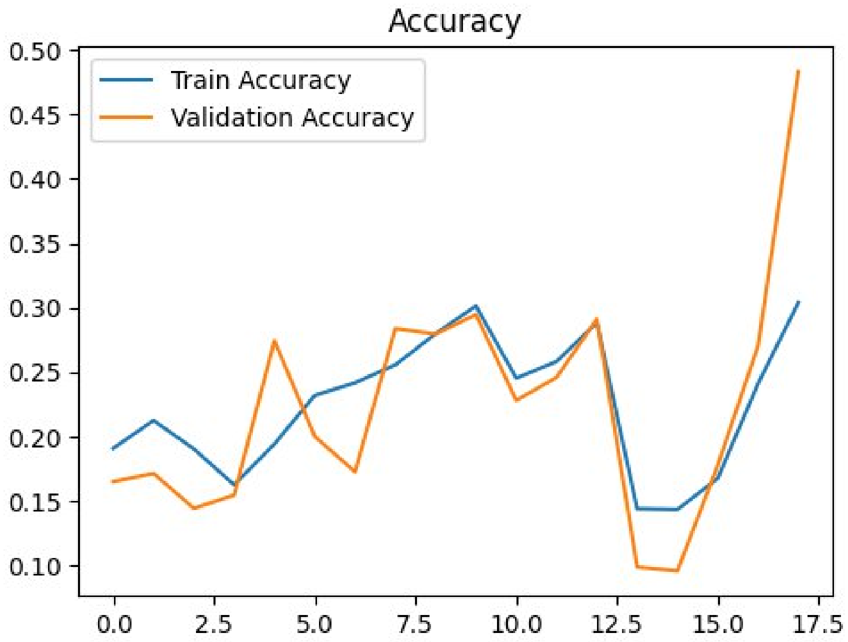


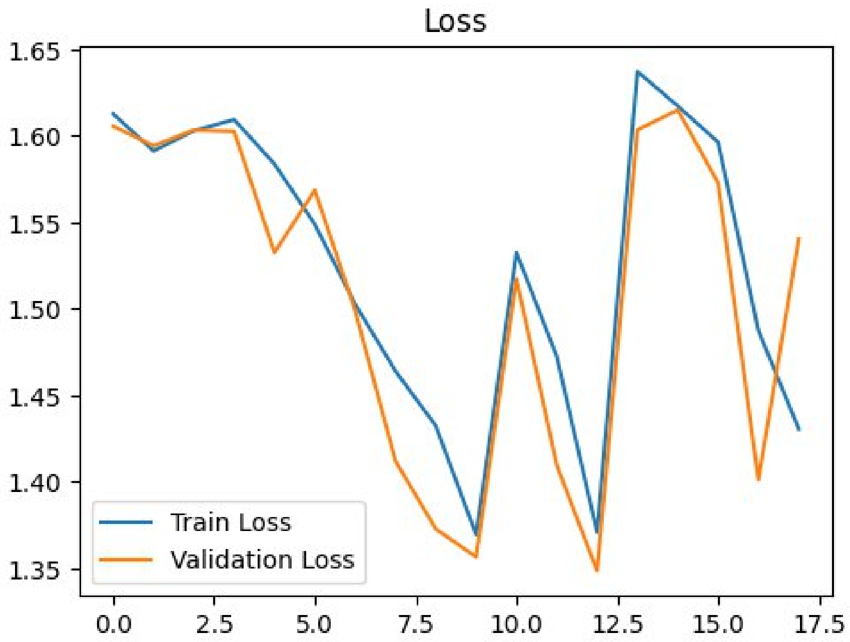
Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 3000, 64)	17,664
dropout (Dropout)	(None, 3000, 64)	0
lstm_1 (LSTM)	(None, 32)	12,416
dense (Dense)	(None, 32)	1,056
dense_1 (Dense)	(None, 5)	165

IMPORTANT NOTE

In this case we are not using extracted features! Extracted features already simplify and summarize the EEG signals. **Using an RNN on these extracted features may not fully exploit its strengths** (direct application of RNNs to raw EEG signals might yield better results in some cases).

Model-4: Results





The RNN model performed poorly, with accuracy and loss graphs showing no clear learning curve. Fluctuations in validation accuracy and loss suggest the **model failed to learn meaningful patterns**, for this reason **this model was also scrapped**.

Comparing this to the SVM, which achieved 70% accuracy using frequency-domain features, highlights the importance of feature engineering. Frequency-based features capture key physiological patterns, making classification easier for SVMs, which work well with structured, low-dimensional data. In contrast, the RNN struggled with raw EEG data, likely due to its high dimensionality and the need for more data and computational power.

Model-5: Chambon Et Al CNN on Raw Data

To achieve better results, we need a network specifically designed for sleep stage classification. An online research suggested us that the sleep staging architecture by Chambon et al. (2018), adapted by Banville et al. (2020), could be a strong candidate. This specialized CNN is designed to classify raw EEG signals effectively.

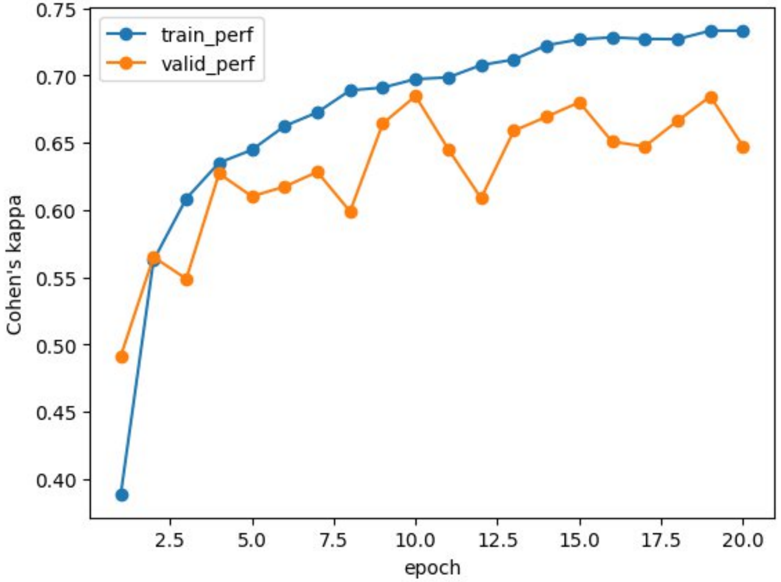
This is a **convolutional neural network** (made using keras library) where:

Layers	Summary
The input:	is a 30-s window of 4 channels.
The hidden layers:	there is a succession of convolutional layers, max pooling, and nonlinearities (Relu). In the end feature maps are flattened and passed through a fully-connected layer.
The output:	is a 5-dimensional vector where each dimension is matched to one of our 5 classes (Wake, N1, N2, N3 and REM sleep stages).

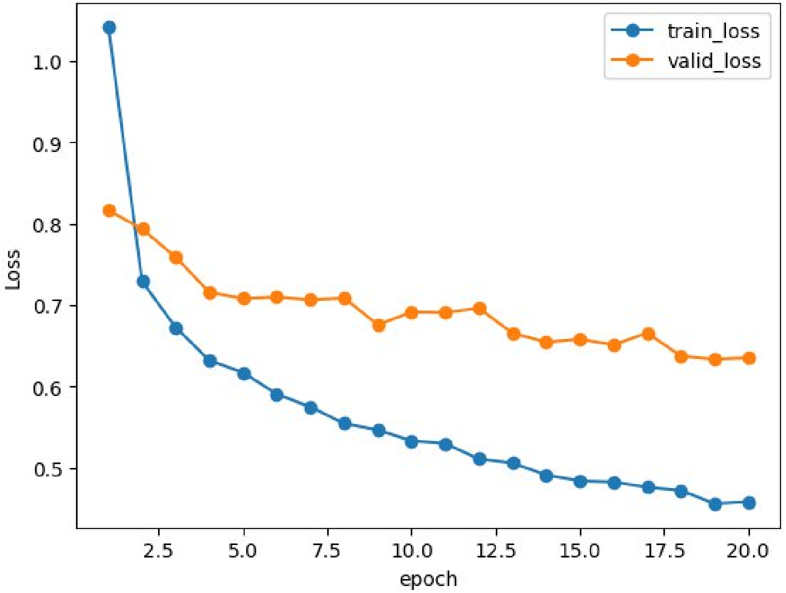
Please note that during each epoh of training, instead of accuracy we mesure **Cohen's kappa**. This score is a statistical measure of inter-rater agreement or inter-annotator agreement for qualitative (categorical) data. It is commonly used to evaluate the performance of classification models, especially in **cases where the classes are imbalanced**.

Model-5: Result (part 1)

Train performance:



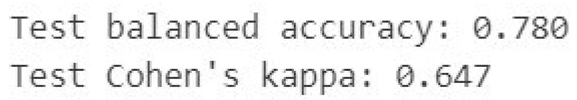
Train loss:



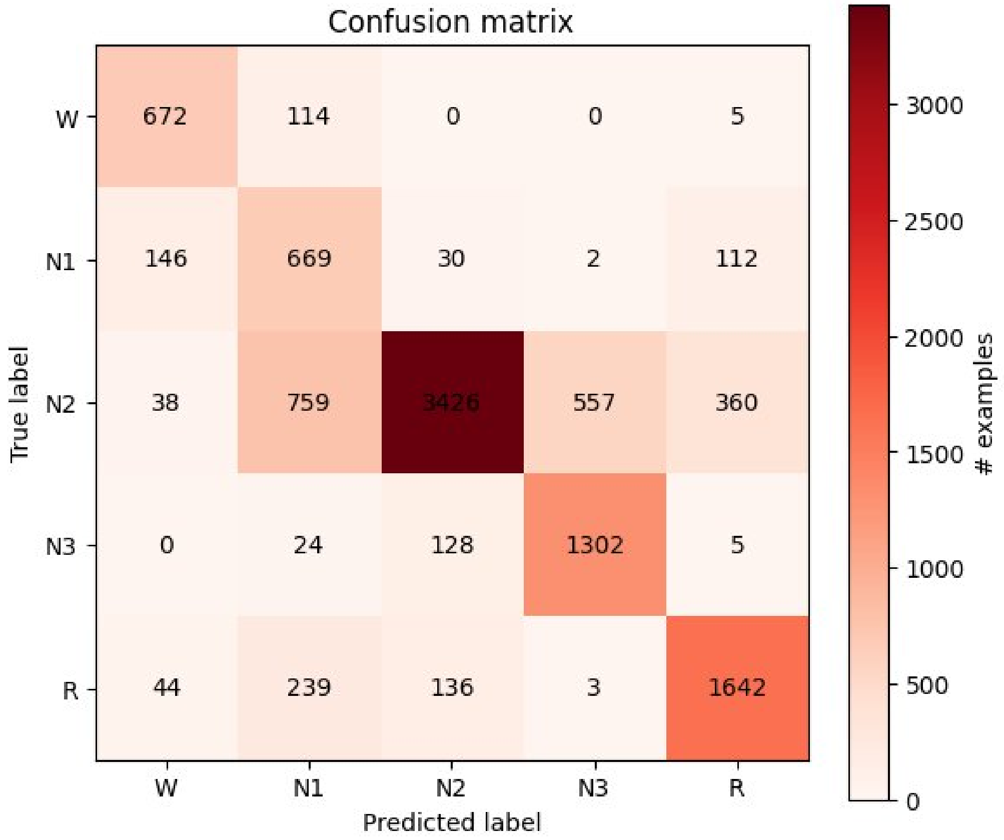
The Chambon et al. model shows steady improvement, with loss decreasing and **Cohen's kappa increasing consistently across epochs**. This indicates successful training.

The somewhat close alignment of training and validation curves suggests **minimal overfitting**. **Fluctuations in Cohen's kappa are to be expected** due to its calculation method, but as long as it trends upward in both train and validation, it's a positive sign.

Model-5: Results (part 2)



```
Test balanced accuracy: 0.780
Test Cohen's kappa: 0.647
```



Confusion matrix

True label \ Predicted label	W	N1	N2	N3	R
W	672	114	0	0	5
N1	146	669	30	2	112
N2	38	759	3426	557	360
N3	0	24	128	1302	5
R	44	239	136	3	1642

We've finally achieved a solid accuracy, approaching 80%.

The confusion matrix closely resembles the one from attempts 2 and 3 using frequency-domain features. Specifically:

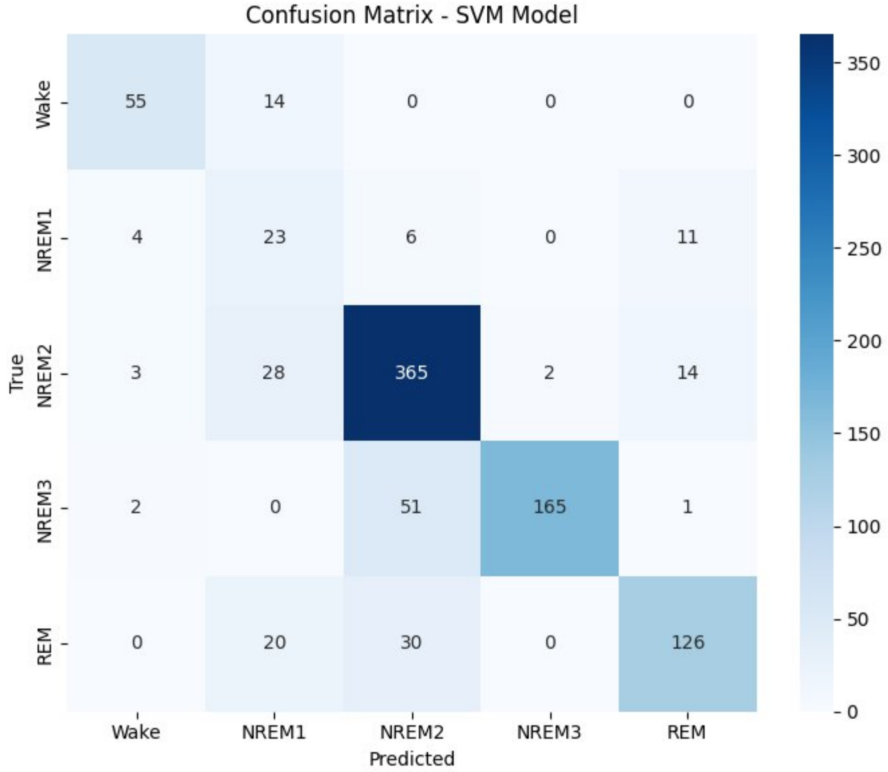
- **The N2 class** (the oversampled one) remains the most accurately predicted.
- **Classes below N2** also show strong performance, with similar true positive counts.
- **Classes above N2** are more challenging to classify, (yet looks like the model did a better performance classifying those than SVM)

These findings align perfectly with our observations from previous models 2 and 3

Best Models Comparaison

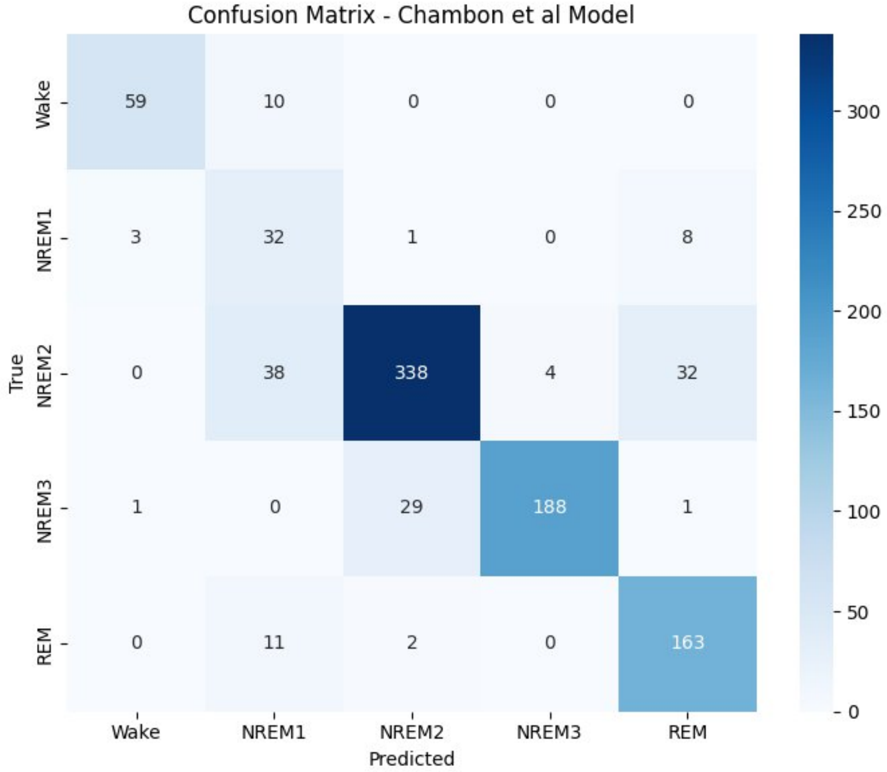
Accuracy Comparaison

Here we make a comparaison between model-3 and model-5 as they are the ones with best overall results



Confusion Matrix - SVM Model

True \ Predicted	Wake	NREM1	NREM2	NREM3	REM
Wake	55	14	0	0	0
NREM1	4	23	6	0	11
NREM2	3	28	365	2	14
NREM3	2	0	51	165	1
REM	0	20	30	0	126



Confusion Matrix - Chambon et al Model

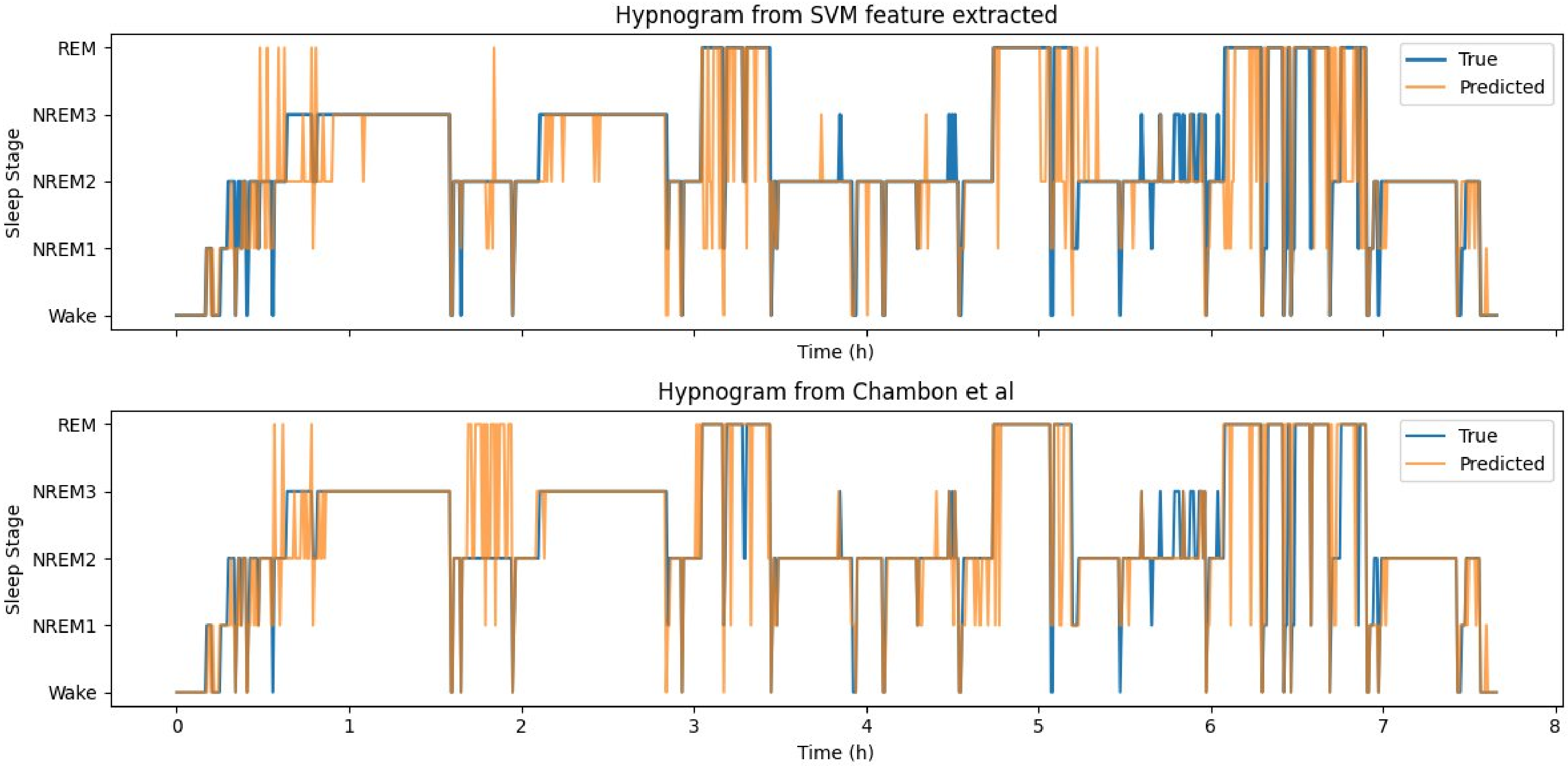
True \ Predicted	Wake	NREM1	NREM2	NREM3	REM
Wake	59	10	0	0	0
NREM1	3	32	1	0	8
NREM2	0	38	338	4	32
NREM3	1	0	29	188	1
REM	0	11	2	0	163

The confusion matrices from both models align well with true sleep stages, supporting our accuracy results. The similarity between the plots reinforces the idea that both models produce comparable classifiers. Key differences include:

- **SVM** **is more precise in classifying REM sleep**, likely due to Power Spectral Density effectively capturing its distinct features.
- **Chambon et al.** **is better at identifying wake stages**, which makes sense since wakefulness is marked by high-frequency activity, and our feature extraction didn't emphasize high frequencies.

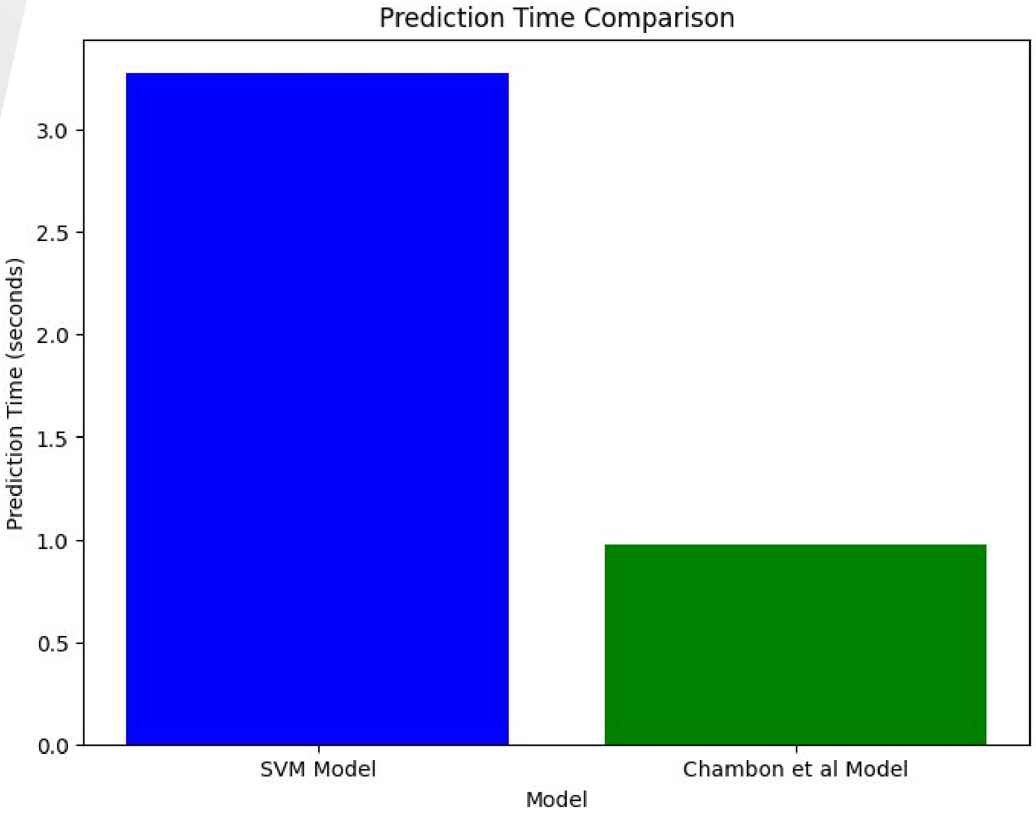
Accuracy Comparaison

Same conlcusions can be deducted from prediction hypnograms comparaisons:



Times Comparaison

Here we make a comparaison between model-3 and model-5 on training and prediction time.



Another intrsting fact is that chambon et al model takes much longer times to train than a simple SVM:

SVM	Chambon
2 seconds	20 minutes

Yet, wen it comes down to prediction time (as shown in graph in the left) Chambon et al is much faster, as it doesn’t need to lose time manually extracting features

This makses the two models both valuable candidates for diffent usecases

Conclusions

Our results demonstrate that effective feature extraction allows a simple SVM to achieve classification performance comparable to specialized deep learning models like Chambon et al.'s. This **highlights the importance of domain knowledge in feature engineering** and the efficiency of classical machine learning.

Key Insights:

- **Efficiency of Simpler Models:** SVM with extracted features required just 1–2 seconds for training, making it ideal for resource-constrained environments.
- **Higher Cost of Deep Learning:** Chambon et al.'s model, which processes raw EEG data directly, took ~20 minutes to train due to its complexity and data demands. Yet this is complimented by receiving better perfomances overall.

Practical Implications:

- Classical models are well-suited for rapid prototyping and low-resource settings.
- Deep learning models are preferable for high-stakes tasks where accuracy justifies higher computational costs.

Overall, choosing between these approaches depends on the trade-off between performance, efficiency, and resource availability.