



Neurobit Technologies
unlock your potential

z3score
.com
speed • accuracy • reliability

Getting started with z3score real time module

Real-time sleep staging and intervention



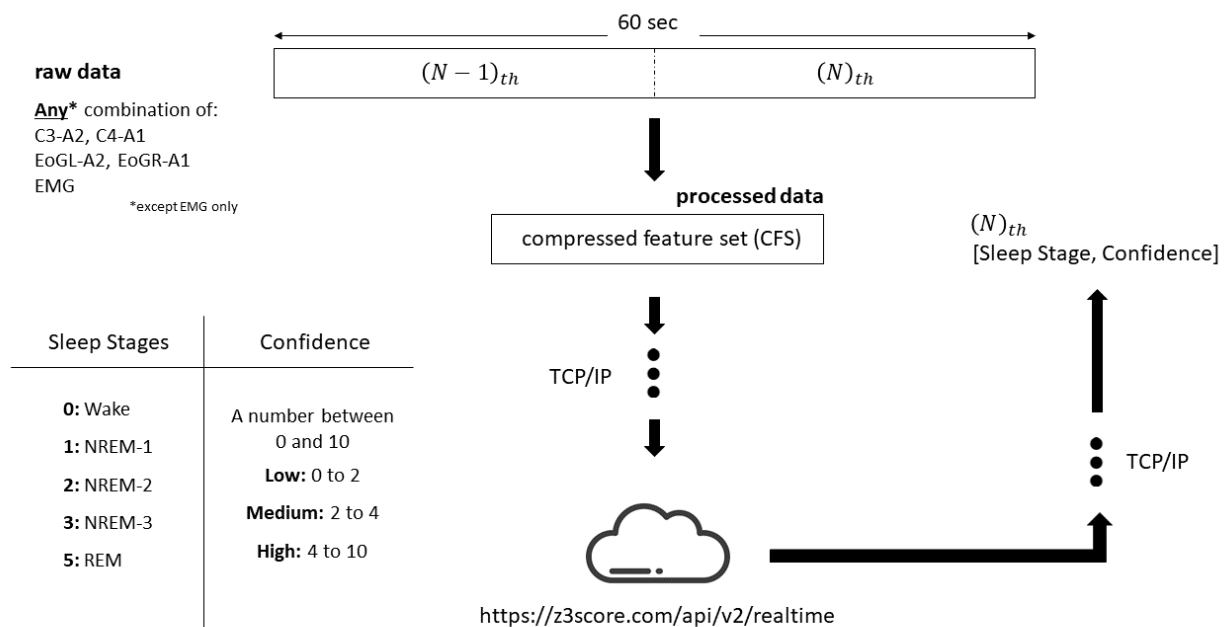
Overview:

Z3Score provides a cloud-based architecture to carry out low latency real time sleep staging over the cloud to design cutting edge intervention and neurofeedback experiments and products. Potential applications include:

- Closed loop stimulations, including auditory, tactile, tACS and tDCS
- Targeted memory reactivation
- Lucid dreams
- Improving behavioural inhibition in ADHD

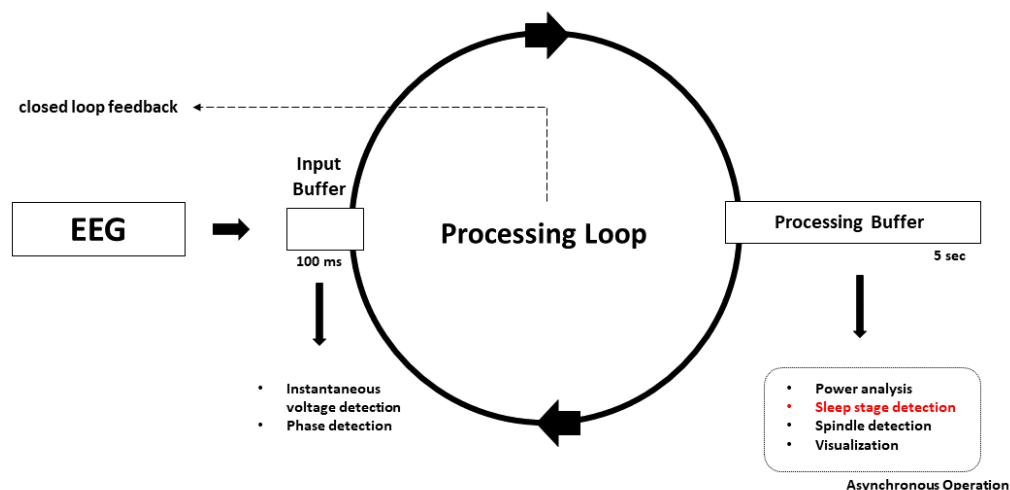
Advantages of the Z3Score cloud system:

Z3Score provides low cost, highly accessible and scalable framework for real-time sleep staging. As the bulk of the most intensive computations are carried out in the server, low cost mobile systems can take advantage of Z3Score's accurate and reliable analytics. Additionally, Z3Score allows staging using reduced channels, including just single EoG channel without significant effect on accuracy. This allows for faster set-ups and even home-based deployments. Z3Score uses industry standard REST API and is programming language agnostic. Basic architecture of the real-time staging system is shown below:



Basics of real-time processing:

The basics of a real time EEG processing system is shown below. The data from the EEG device is collected and sent over to the client. The data is usually stored in an input buffer, the size of which depends on the speed of feedback required. The input data is then collected and processed in a loop. This loop must run fast enough that it completes all analysis before the input buffer fills up. Usually, the data being collected is accumulated in a processing buffer, as some processing requires a minimum amount of data. If some of the processing is CPU intensive, it might be necessary to run them asynchronously to avoid slowing down the processing loop.



In the example shown above, the input buffer is of size 100 ms. Therefore, the feedback is always delayed by at least 100 ms. The processing loop must execute within 100 ms to avoid buffer overflow.

Getting started:

We recommend installing Python 3.6 or higher.

- **demo_basic.py** Start with this example to understand the basics
- **demo_tmr.py** This is a realistic example to demonstrate TMR using our platform.

For more information contact sales at sales@neurobit.io
the information presented here are subject to change without notice.