

Selene

**An android application for personalized
mental tiredness monitoring using EEG**

Selene Team



Andrea

Model training and evaluation, noise reduction, features extraction and selection, mental workload monitoring logic



Carlo

AI Design, Model training, Feature extraction, Signal processing, Model personalization



Dawid

Data reading, saving, collection and analysis



Lorenzo

UI/UX Design, Tracking data from EEG

Overview (and the *challenges*)

- In recent years, interest in mental well-being has grown, along with the high cognitive demands placed on students and professionals
- Prolonged mental effort can harm and health, making necessary real-time, portable tools for monitoring cognitive state
- Many current EEG-based solutions are generic and **fail to account for individual variability**



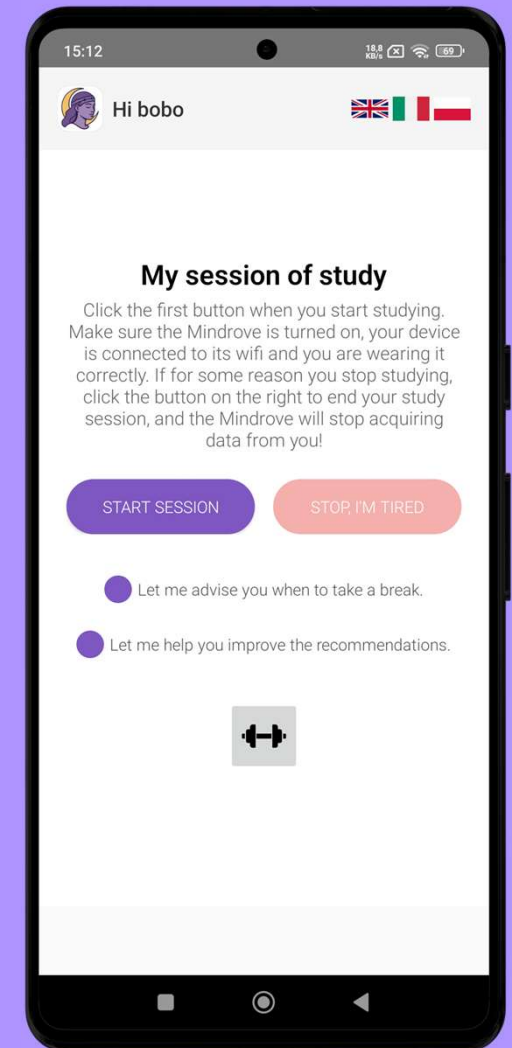
MindRove Arc: Our Sensor

- EEG electrode locations: C5, C3, C1, C2, C4, C6
- Sampling rate: 500 Hz
- Compatible with Android Studio

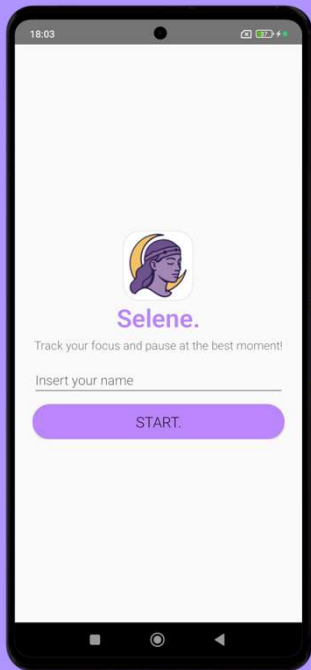


What's Selene

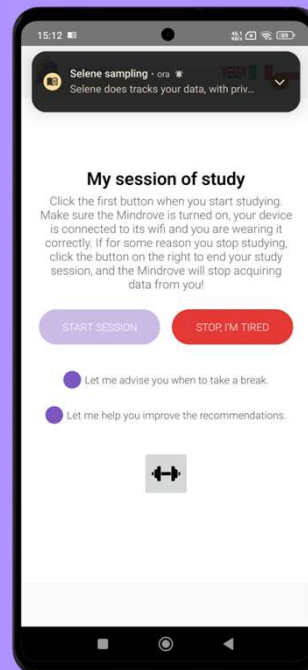
- Android application that advises you when to take a break from studying.
- It uses a customized AI model based on the EEG signals received from the user.
- User privacy is ensured.



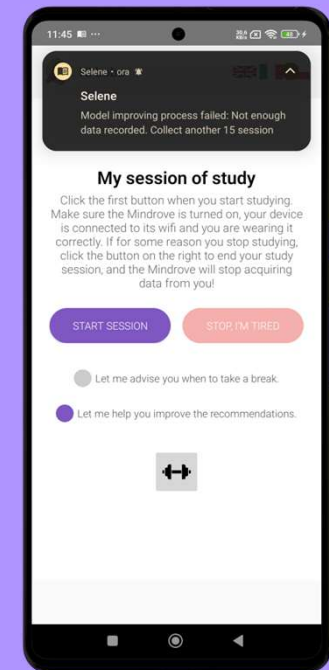
Application Workflow



1. Starting page



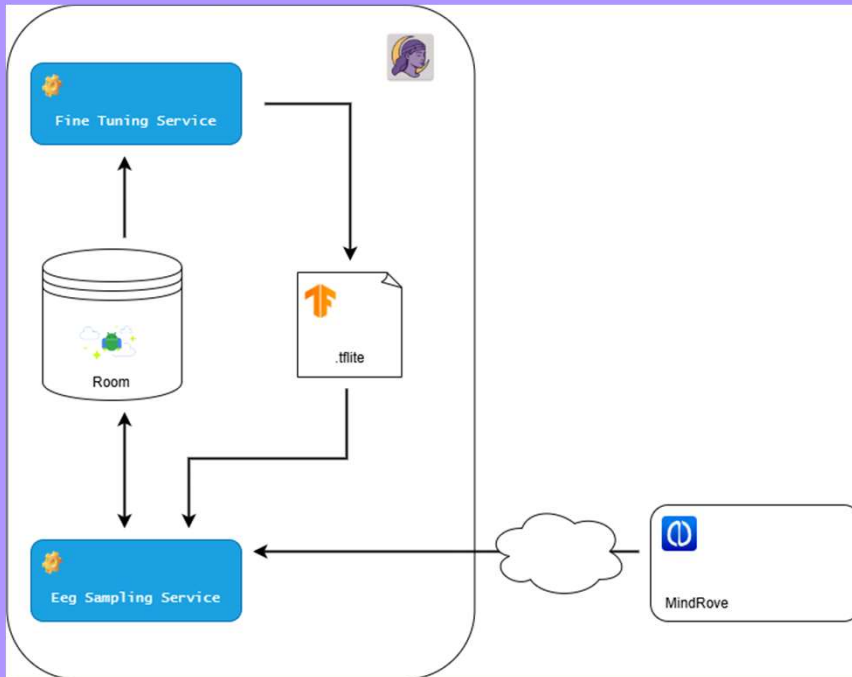
2. Starting a session



3. Training the model



Application Architecture



Tensor Flow Lite

TensorFlow Lite is a lightweight version of TensorFlow designed to run machine learning models on mobile devices



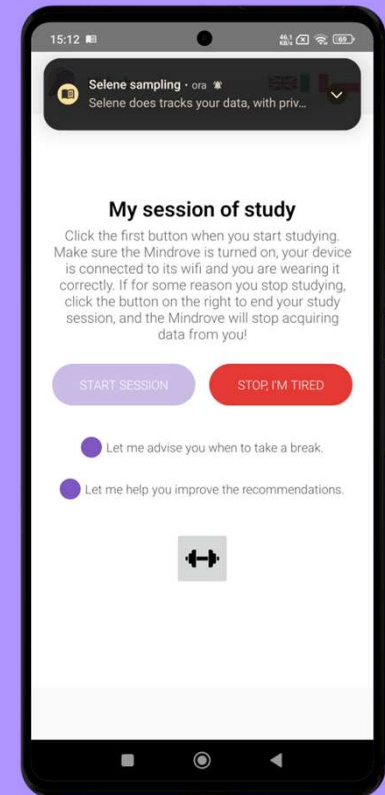
Room

Room is an Android persistence library that provides a level of abstraction over SQLite

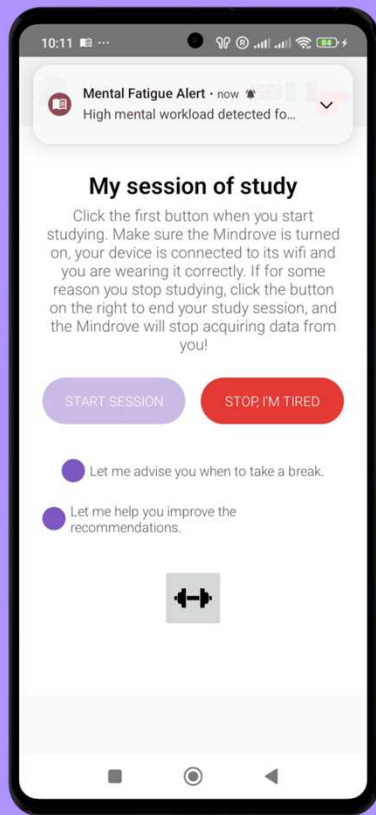


EEG Sampling Service (1)

- The Server Manager of the MindRove library starts a listener that receives packets of SensorData in real time (500 Hz).
- When the list reaches 250 samples, the data is written in batches in the SampleEeg table to improve efficiency.
- The level of tiredness is applied to EEG data collected in the last 16000 samples (32 seconds) stored in the SampleEeg table.



EEG Sampling Service (2)



- The last 16,000 EEG samples are retrieved, preprocessed, reduced to 100 Hz (3,200 samples), features are extracted, and the fatigue level is classified (from 0 to 3) using a neural network.
- Each prediction is saved in a buffer (maximum 18 values). Once the buffer is full, the mode is calculated to estimate the fatigue level.
- If the level is ≥ 2 , a break is recommended; otherwise, you are encouraged to continue. After the notification, the buffer is emptied; otherwise, only the oldest value is removed.



Baseline Model Training

Model Type:

→ Feed-Forward Neural Network (FNN)

Reason for selection:

→ Similar performance to XGBoost, Decision Trees, Random Forest

→ Best compatibility with TensorFlow Lite



Choosing TensorFlow Lite

Why TensorFlow Lite:

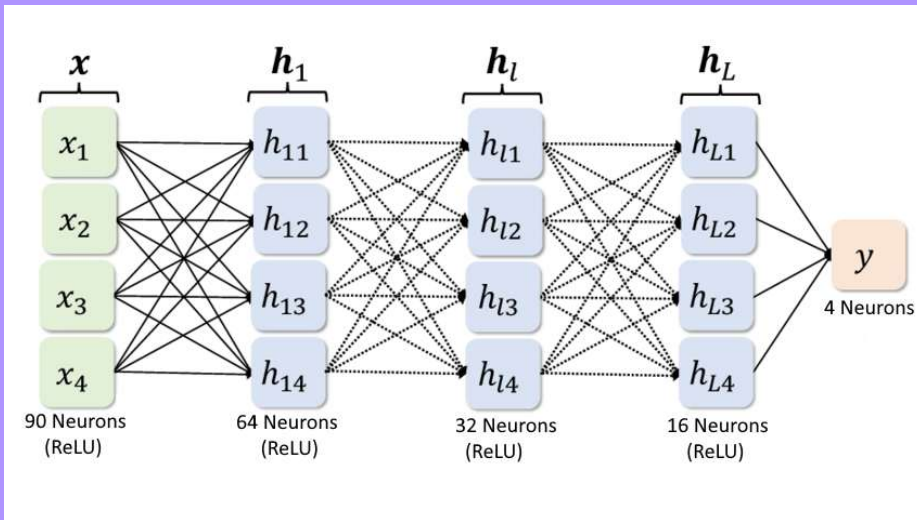
- Optimized for mobile deployment
- Low inference time and resource consumption
- No need for external server connection

Strategy:

- Model pre-trained externally
- Personalized on-device through fine-tuning



Model Architecture



Model: Feed-Forward Neural Network

Fine-tuning:

→ First 3 layers are frozen

Signatures implemented:

→ Train, Classify, Save, Load_weights

→ Enable training/inference after conversion to .tflite



Training Dataset

Dataset: Public EEG dataset from 40 subjects

Recording specs:

→ 32-channel EEG, 100 Hz

→ 4 cognitive tasks:

→ Mental arithmetic

→ Stroop color-word test

→ Symmetrical image recognition

→ Resting state

Issue:

→ Labels range from 0 to 10 (imbalanced distribution)



Label Reorganization

From 11 fatigue levels to 4 classes:

→ **Class 0 (Rest):** levels {0, 1}

→ **Class 1 (Not fatigued):** {2, 3, 4}

→ **Class 2 (Slightly fatigued):** {5, 6}

→ **Class 3 (Fatigued):** {7, 8, 9, 10}

Goal: Balanced class distribution



Preprocessing Pipeline

Objective: Clean EEG signals before feature extraction

Steps:

→ **Trend Removal:**

→ Savitzky-Golay filter (order 5, frame-len 127)

→ **Noise Reduction:**

→ Wavelet thresholding (Daubechies2, soft thresholding)

→ **Notch Filter:**

→ 50 Hz to reduce power line noise



Feature Extraction

Channels used: Central EEG channels

Total features per channel: 29

→ **Time-domain:** MIN, MAX, RMS, VAR, STD, POWER, P2P, SKEW, KURTOSIS, etc.

→ **Frequency-domain (FFT + Welch method):**

→ Absolute and relative power (Delta to Gamma)

→ Alpha/Theta, Theta/Alpha and (Alpha+Theta)/Beta ratios

→ Peak frequency, spectral entropy

Input format: 6×29 matrix per window



Feature Selection

Method:

→ Based on absolute weights in the layers

Process:

→ Sum of absolute weights per feature-channel pair

→ Aggregated per feature across channels

→ Top 15 features selected

Final input:

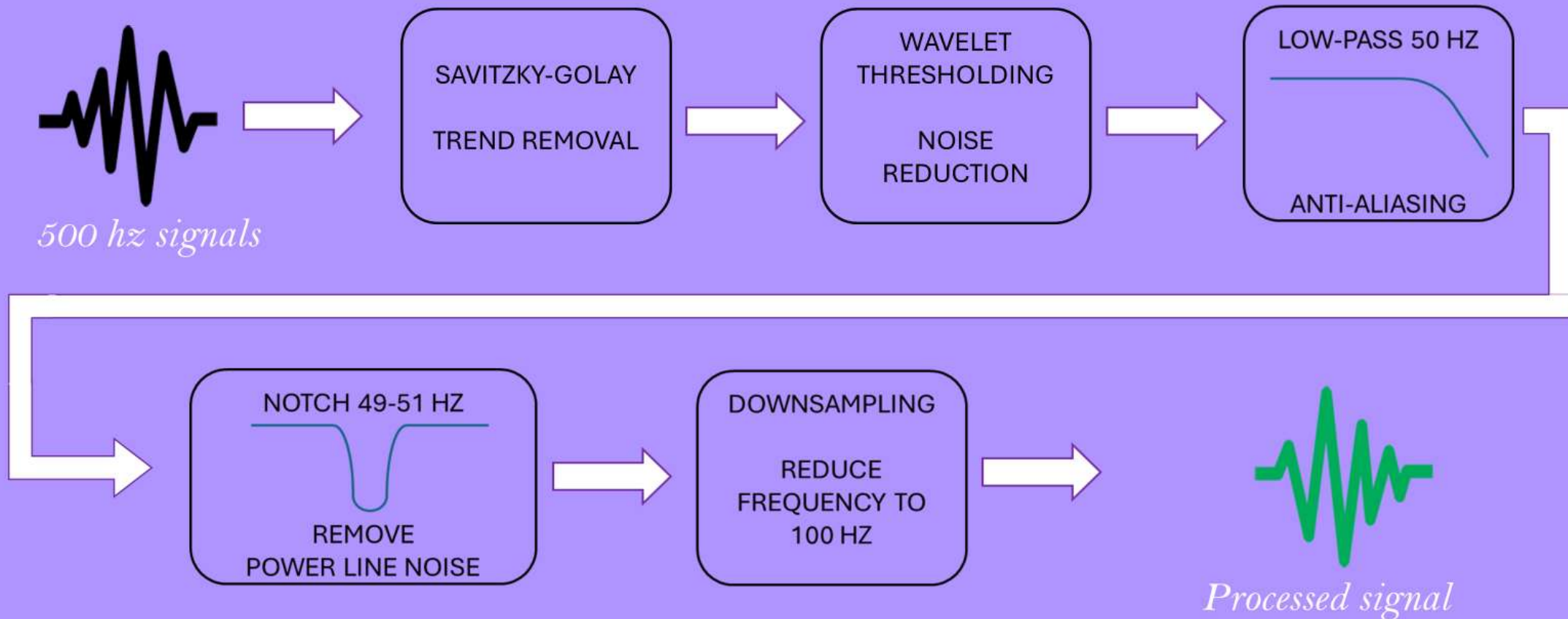
→ 6×15 matrix (15 most relevant features \times 6 channels)

Selected features include:

→ Abs beta Power, RMS, Power, Theta to Alpha Ratio, Rel delta Power, VAR, Rel theta Power, FORM FACTOR, Abs theta Power, Abs alpha Power, PULSE INDICATOR, Spectral Entropy, Rel alpha Power, Theta Alpha to Beta Ratio, Abs delta Power



On device signal processing pipeline

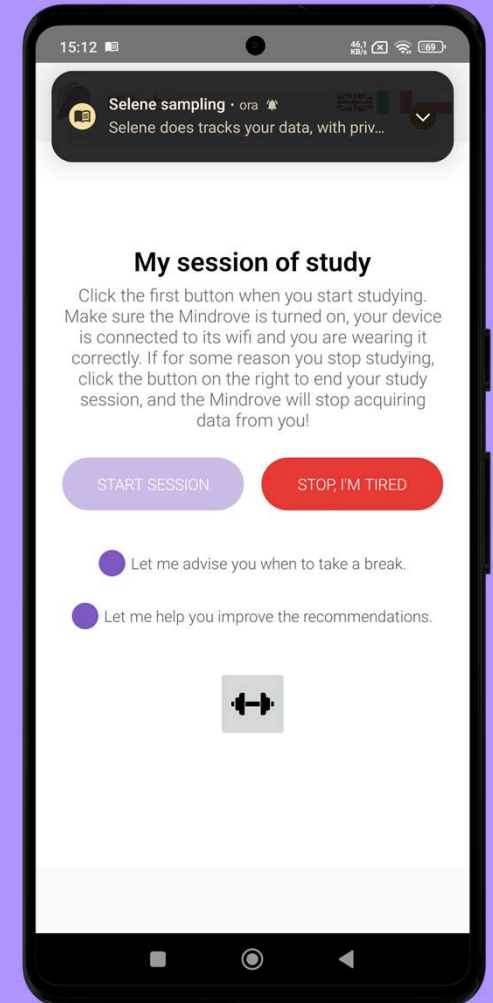


Model personalization: How it works

The base model, use 4 custom methods called "signatures":

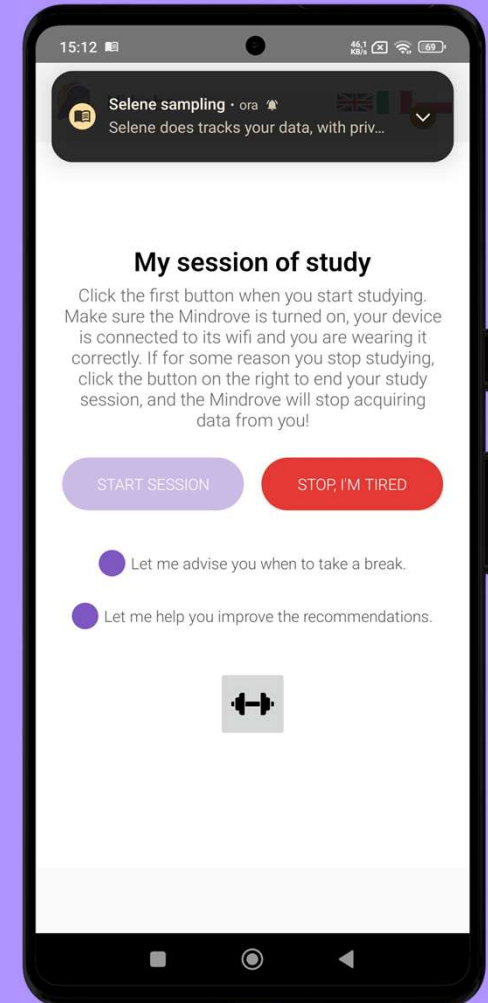
- Train, allows to train the model
- Save, for saving the current weights of the model in a file
- Classify, given an input, classify the mental fatigue
- Load_weights: allows to load weights from a file

Why using signatures? : Once converted using TFlite, the signatures are embedded in the model, allowing to be called on device.



Model personalization: Data Acquisition

- User can assign a level of tiredness at the end of each study session
- The last 32 seconds of a session are stored
- When at least 20 sessions are collected the user can run the personalization of the model



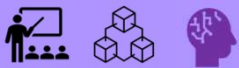
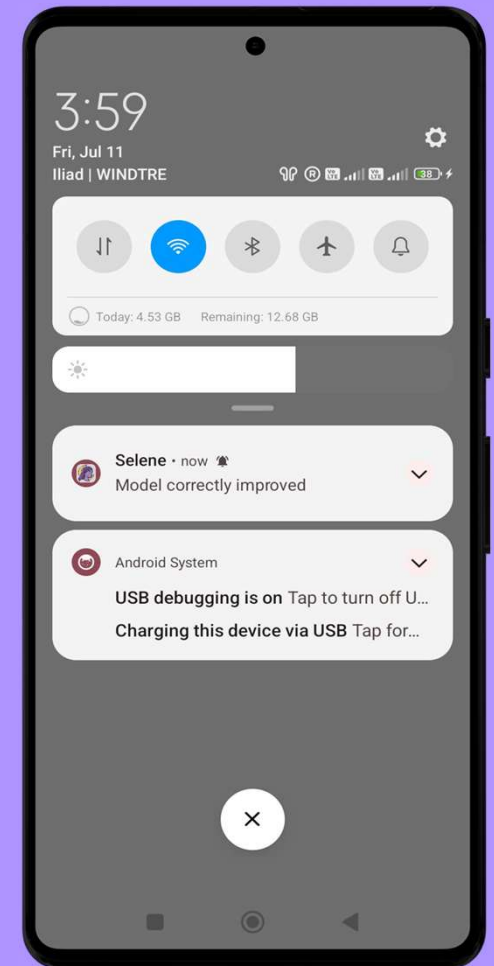
Model personalization: Details

- Head, first hidden layer e second hidden layer are frozen
- Online training (batch size equal to 1)
- Epochs performed 20

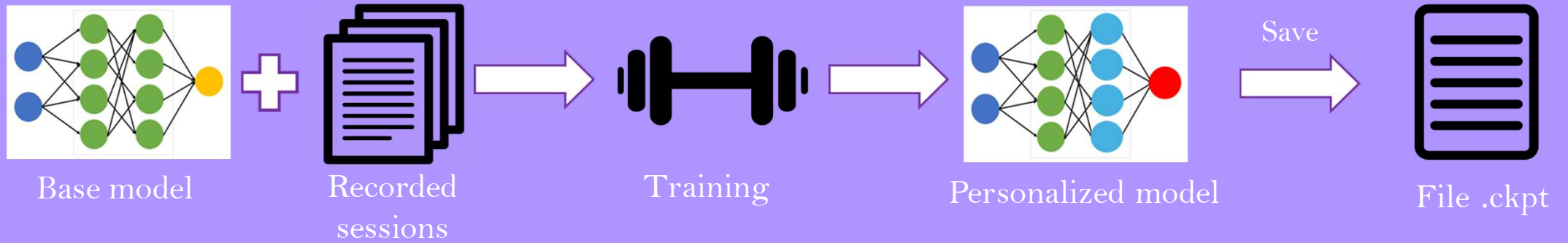
Why this choices? Resource constrained environment, energy constraints

At the end of the training, the model save the updated weights in a checkpoint file.

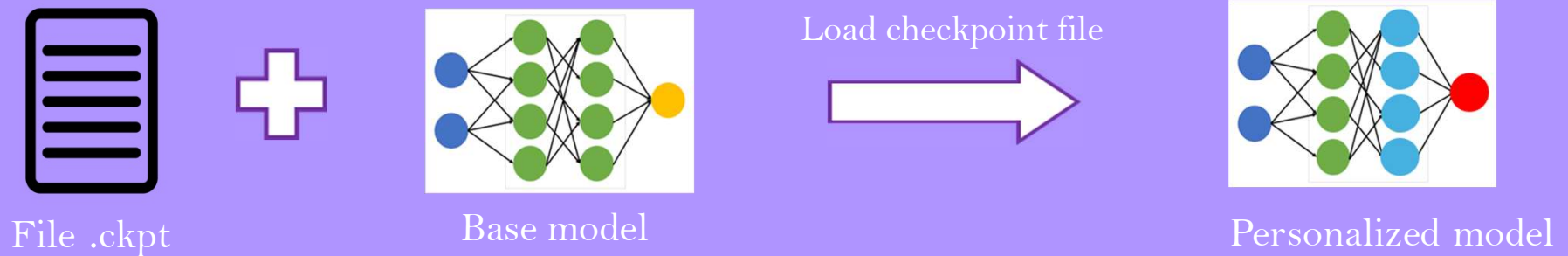
The sessions used to personalize the model are deleted.



Model personalization: Checkpoint files mechanism

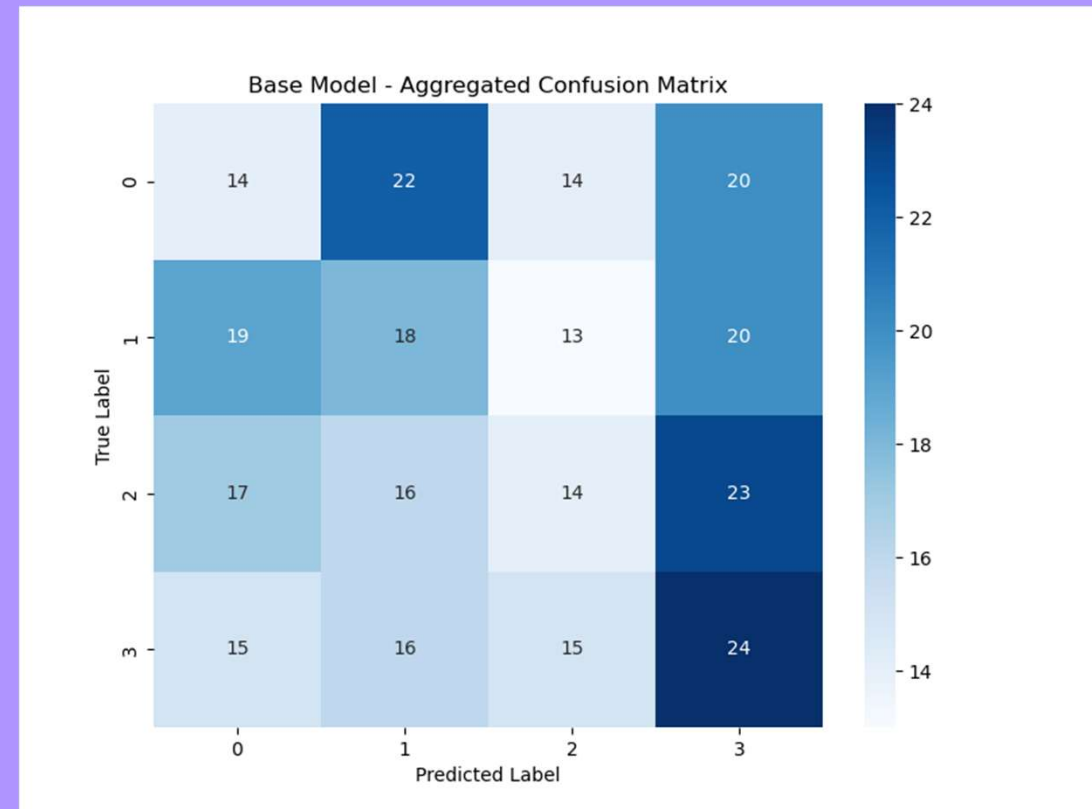


... After the first personalization



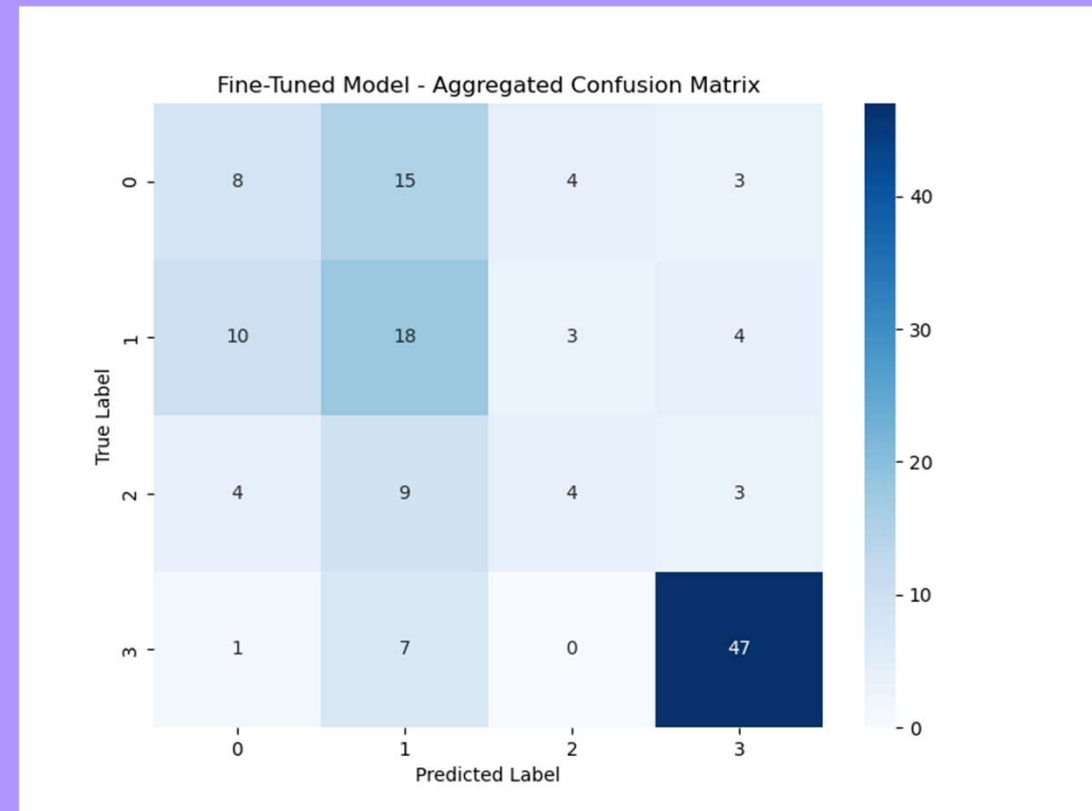
Base model

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.22	0.20	0.21	70.00
1	0.25	0.26	0.25	70.00
2	0.25	0.20	0.22	70.00
3	0.28	0.34	0.31	70.00
macro avg	0.25	0.25	0.25	280.00
weighted avg	0.25	0.25	0.25	280.00



Fine tuned-model

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.35	0.27	0.30	30.00
1	0.37	0.51	0.43	35.00
2	0.36	0.20	0.26	20.00
3	0.82	0.85	0.84	55.00
macro avg	0.48	0.46	0.46	140.00
weighted avg	0.54	0.55	0.54	140.00



Performance comparison

TRUE CLASS	AVERAGE PREDICTED CLASS
0	1.07
1	1.03
2	1.30
3	2.69



Limitations and future work

Limitations

- Base model trained on a small dataset
- Evaluation conducted on one subject
- Data volume dependent performance
- Model biased towards more frequent fatigue levels

Future work

- Collecting the data larger and more diverse group of people
- Considering a more precise EEG device, using more electrodes and BLE communication for energy efficiency
- Using other sensors like accelerometer to detect state of the user or correlating study times with the time of day



Conclusions

- Developed a mental workload personalized user prediction application in 3 languages
- The model learns on the study sessions of the user
- The mental fatigue level is assessed on a scale 0-3
- The prediction service informs the user when to take a break
- Achieved a 54% classification weighted average f1-score after 20 sessions of study

