An Android application for personalized mental

tiredness monitoring through EEG sensor and

incremental learning

Andrea Zanin, Carlo Pio Pace, Dawid Stecko, Lorenzo Valtriani  
 a.zanin@studenti.unipi.it, c.pace10@studenti.unipi.it, [d.stecko@studenti.unipi.it](mailto:d.stecko@studenti.unipi.it), l.valtriani2@studenti.unipi.it

ABSTRACT

The increasing cognitive demands in academic and professional settings necessitate effective tools for managing mental fatigue. This work presents Selene, an Android application that leverages the smartphone as a mobile sensing platform to monitor cognitive workload in real-time. By analyzing electroencephalogram (EEG) signals from a wearable MindRove Arc headset, Selene employs on-device machine learning with TensorFlow Lite to assess a user's fatigue level. A key contribution of this work is the implementation of a user-in-the-loop personalization pipeline, grounded in the humans-as-sensors paradigm. The system integrates a machine learning model that evolves over time by incorporating subjective user feedback, thereby adapting to the individual's unique neurophysiological patterns, a crucial step for improving model generalizability [1], [2]. Initial experimental results are promising, demonstrating that this personalization process significantly improves model performance, increasing the F1-score for detecting high fatigue from 0.31 to 0.84. By combining real-time brain signal monitoring with continuous, on-device adaptation, Selene provides a practical and effective approach to supporting cognitive well-being and productivity.

1 Introduction

In recent years, there has been a growing interest in cognitive and mental well-being, driven by the increasing cognitive demands placed on students and professionals alike [3], [4]. Prolonged mental effort, if not properly recognized and managed, can lead to reduced performance, difficulty concentrating, and long-term negative impacts on overall psychological and physical health [5]. In this context, tools capable of monitoring an individual's cognitive state in real-time are particularly valuable, especially when they are portable, customizable, and easy to integrate into everyday life [3], [6].

The availability of low-cost portable electroencephalogram (EEG) devices, such as the MindRove Arc used in this study, has enabled the development of mobile applications focused on brain activity monitoring [7]. However, a significant challenge in this domain is inter-subject variability; mental fatigue is experienced and expressed differently by everyone, which often renders generic, "one-size-fits-all" models ineffective [2], [4]. Many existing solutions provide standardized measurements, overlooking the fact that personalized models, which adapt to an individual's unique neurophysiology, are required for robust and reliable classification [2].

In this work, we introduce Selene, an Android application developed to address this gap. Selene monitors a user's mental fatigue level in real-time by analyzing EEG signals acquired via a MindRove Arc device. Critically, it implements a user-in-the-loop personalization pipeline based on the humans-as-sensors paradigm [2]. By collecting subjective self-assessments at the end of study sessions, Selene progressively fine-tunes a personalized machine learning model on-device. This approach leverages the convenience of the smartphone as a mobile sensing platform while directly tackling the challenge of inter-subject variability, aiming to create a practical and effective tool for cognitive self-regulation.

The central component of the system is the EEG sensor used in the experiments: the MindRove Arc device presented in Figure 1, whose technical specifications are outlined below Figure 2.



Figure 1: MindRove Arc device.

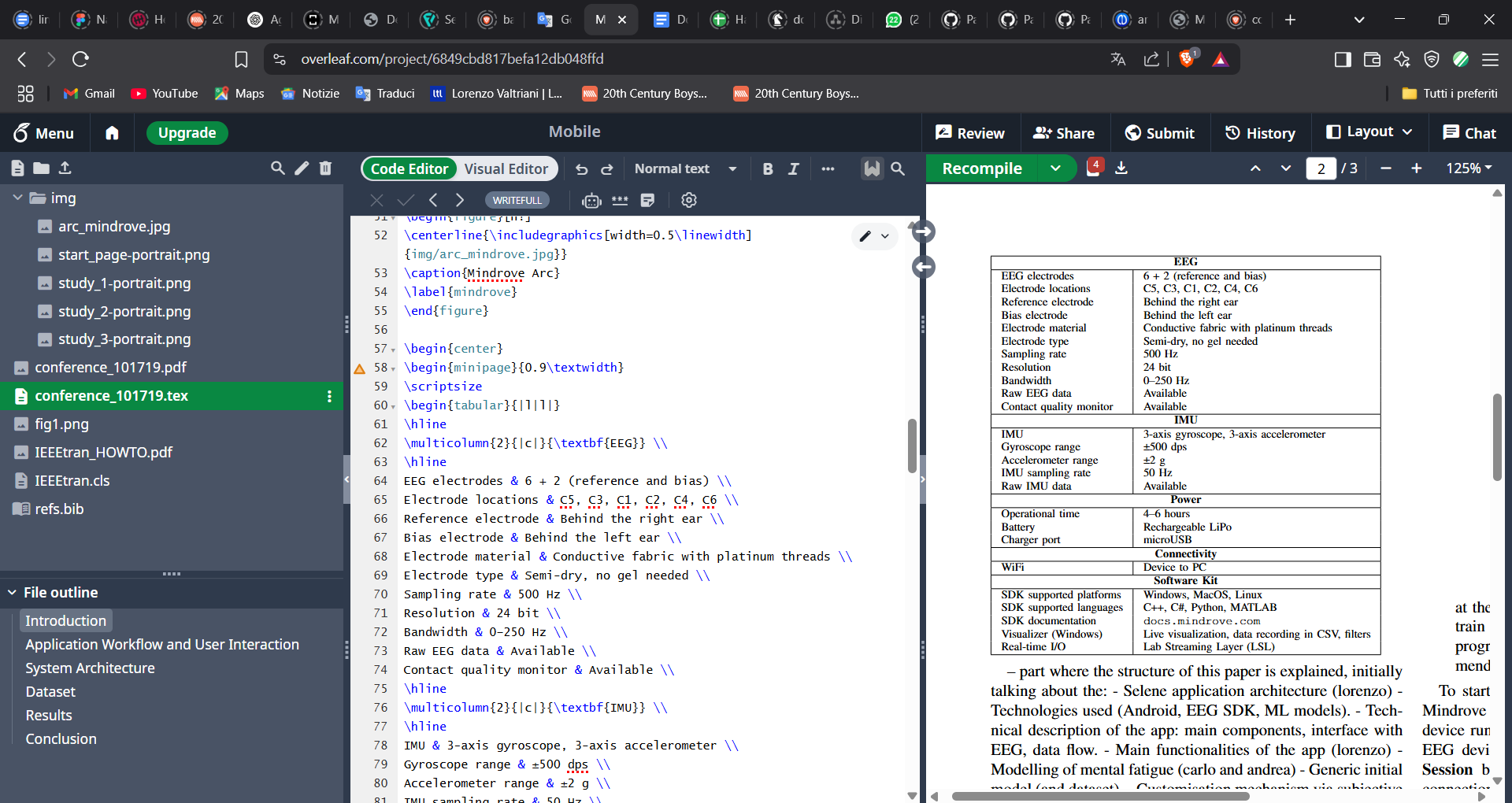


Figure 2: Technical specifications of the MindRove Arc.

2 Application Workflow

This section describes the operational flow of the *Selene application* and the intended usage procedure for users who wish to explore its features. The goal is to provide a clear and practical overview of the steps required for effective interaction with the system.

After installation, the user must ensure that notification permissions are enabled to fully benefit from the app's functionalities. Upon the first launch, the user is prompted to enter a username, which is necessary for user identification within the system.

This initial screen Figure 3 appears only during the first access; in subsequent sessions, the application opens directly on the main interface, labeled *StudySession.*

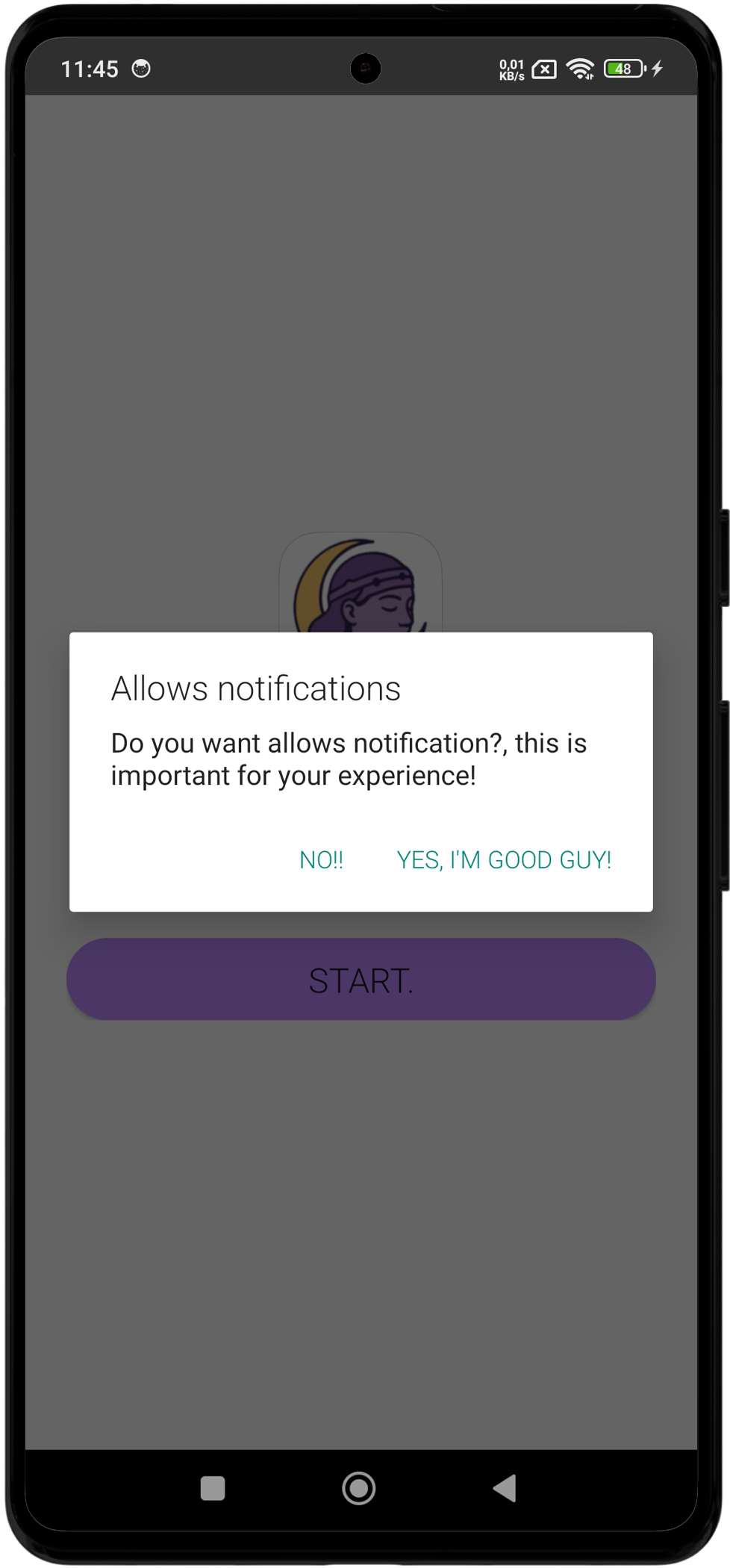


Figure 3: Initial screen of Selene.

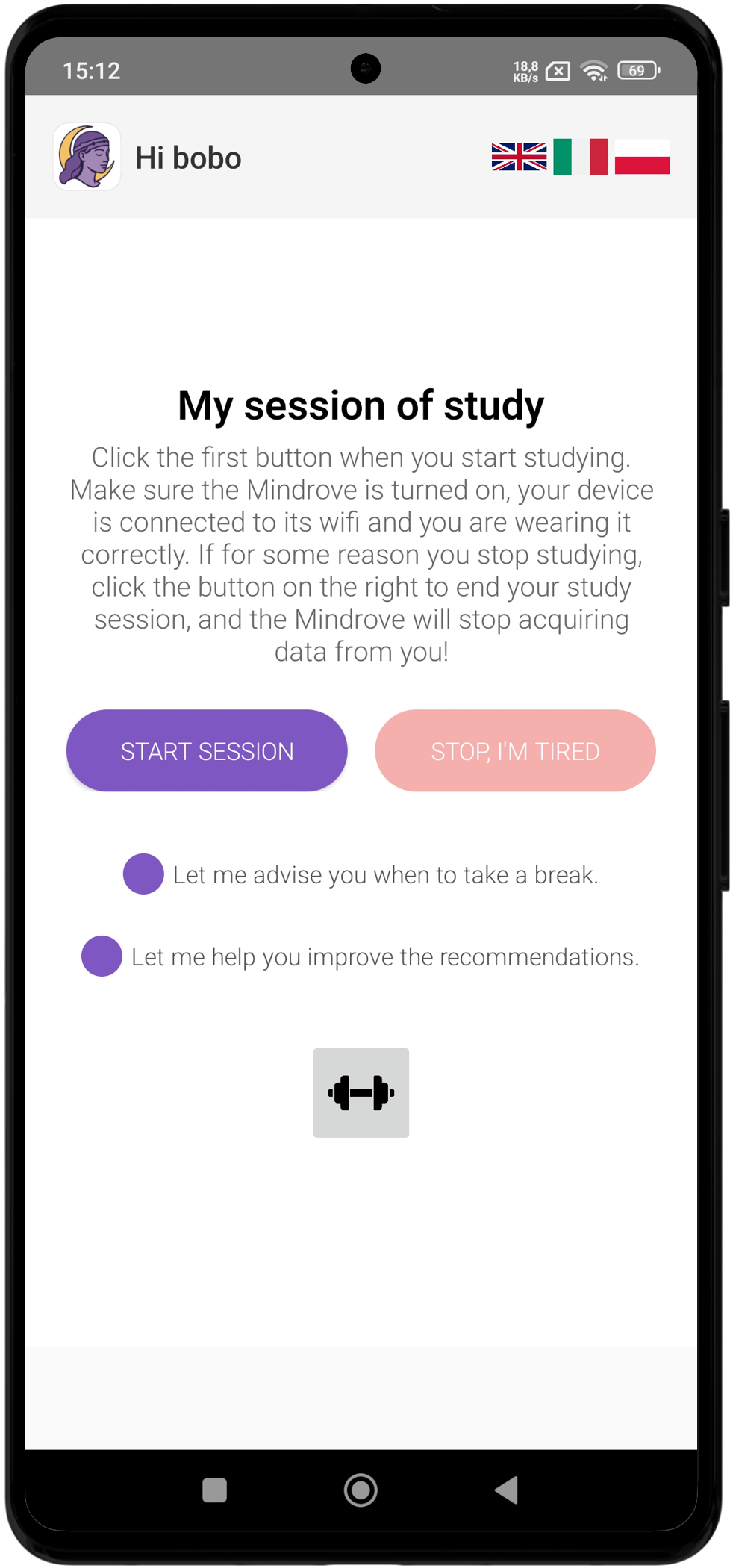


Figure 4: Study Session screen.

The *StudySession* screen

Figure 4 offers the core functionalities, both of which can be activated via checkboxes located at the bottom of the interface:

* *Cognitive Break Suggestions*

When enabled, this feature allows Selene to send notifications recommending a break when a high level of mental fatigue is detected. It also notifies the user when they appear rested enough to resume their activity.

* *Model Personalization*

When enabled, this feature prompts the user to provide a subjective self-assessment of their mental fatigue level at the end of each session. This feedback is crucial for the fine-tuning of the personalized model.

To start a study session, the user must correctly wear the MindRove EEG headset, power it on, and connect the Android device running Selene to the Wi-Fi network generated by the EEG device. Once connected, the user can press the *Start Session* button to initiate the monitoring process. If the connection is active, Figure 5 the app begins receiving and analyzing brain signals in real time otherwise, a notification is shown to remind the user to connect the Android device to the MindRove network.

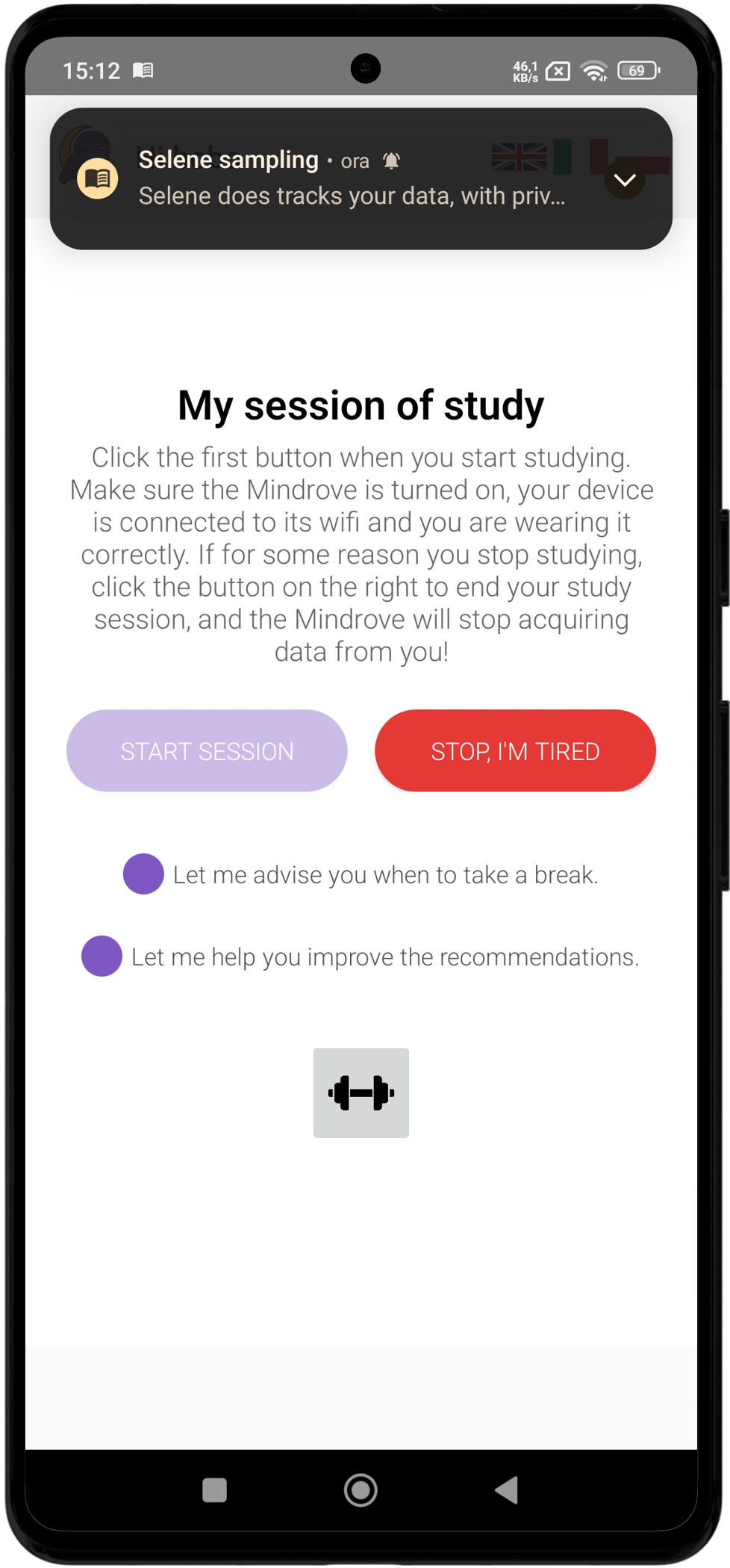


Figure 5: Real time monitoring EEG signals.

During the session, the application can send fatigue alerts even when running in the background, allowing the user to independently decide whether to take a break or continue studying.

At the end of the session, the user presses the *Stop, I’m tired* button, which terminates the EEG connection. If the model personalization option is active, the user is then prompted to select one of four emojis representing increasing levels of mental fatigue Figure 6. This feedback is used to further refine the personalized model stored on the device.

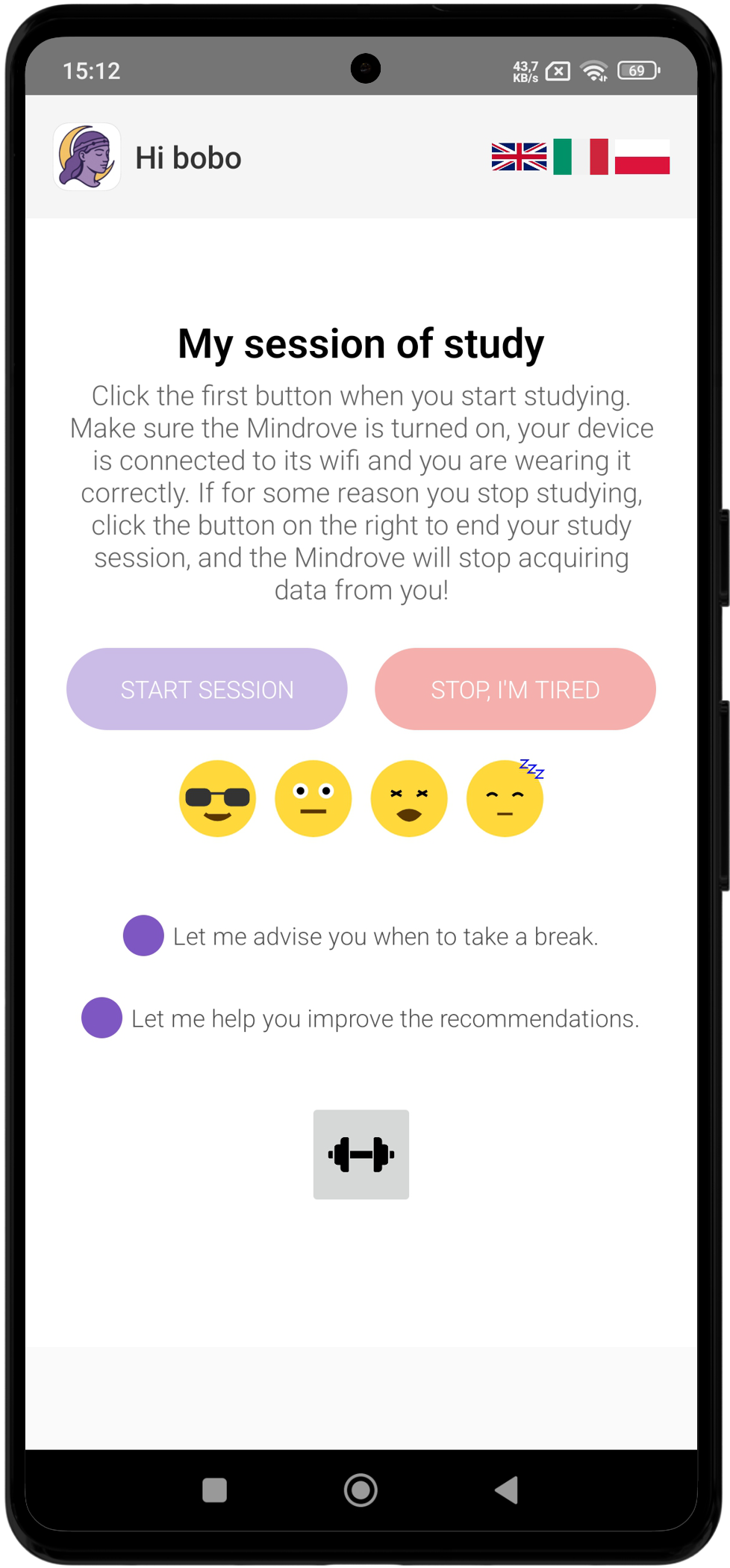


Figure 6: Personalized voting system.

Whenever the user wants, after having contributed enough samples during his study sessions, he can click on the training icon and activate the process, which in the background will help to personalize the model Figure 7.

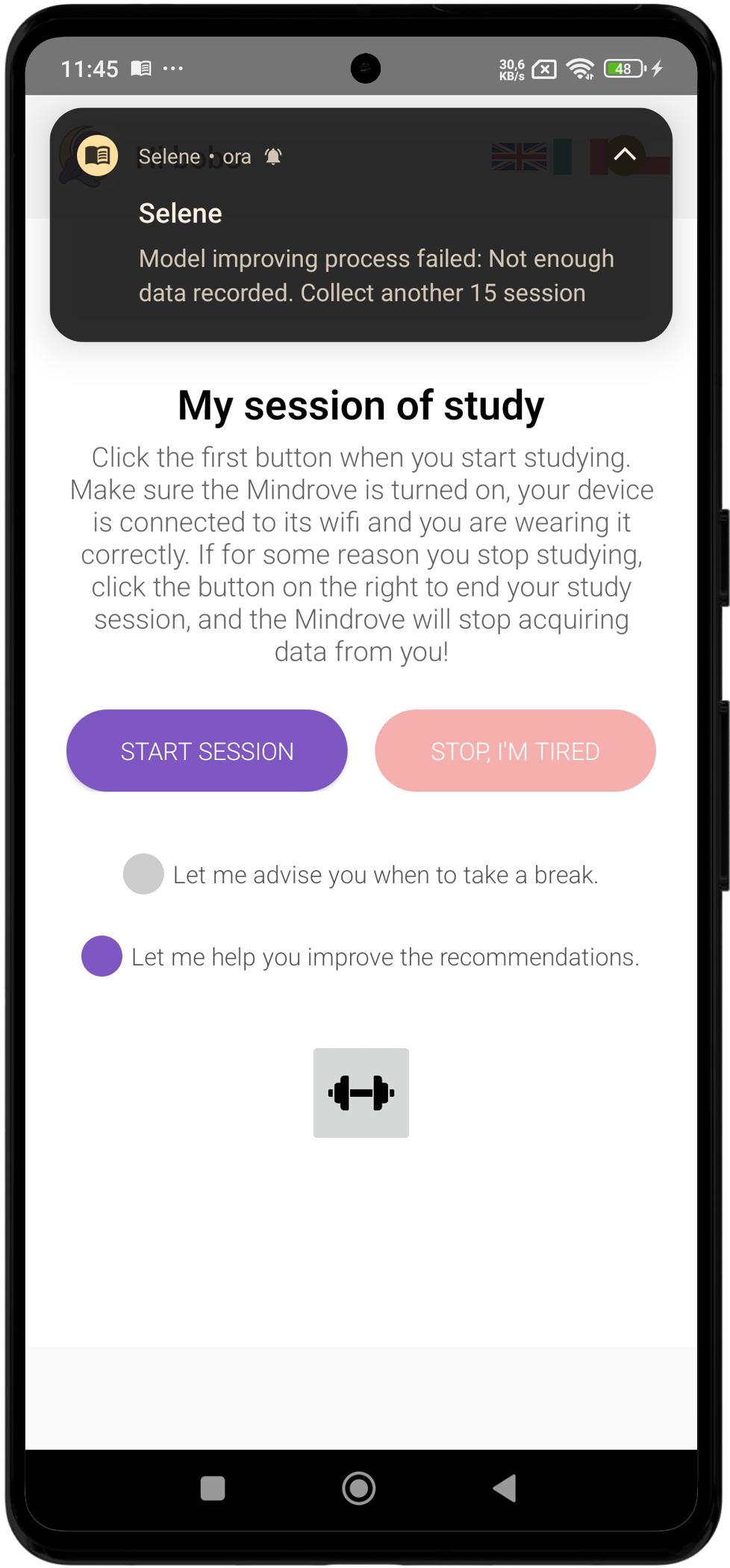


Figure 7: Training personalized model.

The interface displays text in English by default. However, users can switch the language from the settings menu, with support currently available for Italian and Polish.

3 Application Architecture

The system presented in Figure 8 consists of two primary components: an Android application, executed locally on a mobile device, and an EEG device developed by MindRove, which communicates with the Android device via a Wi-Fi connection. The application serves as the main interface for user interaction and manages both computational logic and machine learning processes. Its architecture is centered around two background services, a standard and robust design pattern in the Android framework for managing long-running tasks such as sensor data acquisition and processing without blocking the main user interface thread [6]. This ensures that Selene can provide continuous, real-time monitoring even when the application is not in the foreground, a critical requirement for pervasive sensing systems [6], [8].

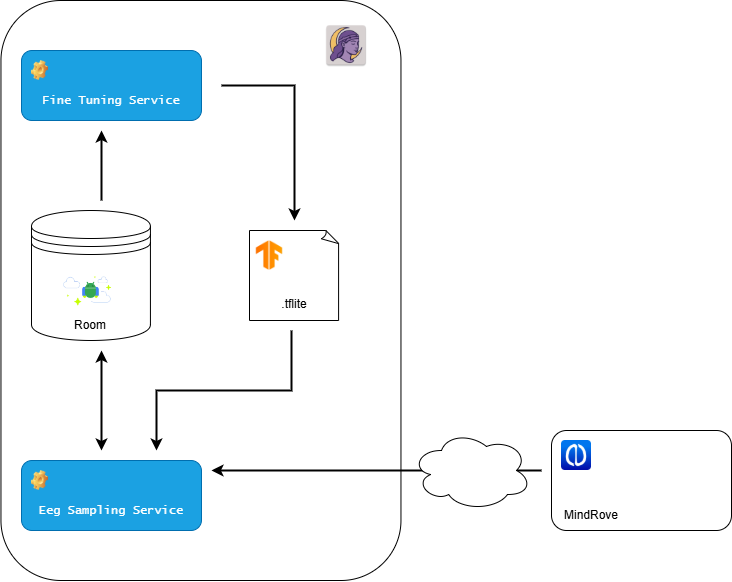


Figure 8: General Architecture of Selene.

3.1 Core Services

3.1.1 EEG Sampling Service

The EEG sampling service is activated exclusively upon successful connection with the MindRove device, a condition that typically occurs during a user’s study session. The ServerManager module, provided by the external mylibrary.MindRove library [9], initiates a listener that receives real-time SensorData packets containing raw EEG signals transmitted by the device.

Within the ServerManager, a lambda function is defined to manage the insertion of EEG data into a local **Room** database. This function is executed at a frequency of 500 Hz, and the incoming samples are first queued in a list rather than being immediately written to the database. Once the list reaches a batch size of 250, the batch of samples is then inserted into the database. This batching strategy reduces the overhead of performing a database write operation for every individual EEG sample received.

At the end of each session, the user is prompted to manually select their perceived level of fatigue. This label is then assigned to all EEG samples collected during the last 16000 samples (last 32 seconds) of the session and is stored on the SampleEeg table.

In parallel, a second component of the service, the MentalWorkloadProcessor class, runs an inference pipeline depicted in Figure 9 at a rate of one operation every 32 seconds. It retrieves the 16,000 most recent stored EEG samples, performs a pre-processing for artifact removal, after applies subsampling to reach 100 Hz frequency, matching the original dataset used to train the model. The resulting 3,200 samples are then processed using time and frequency domain analysis to extract a feature vector, which is fed into a neural network model. The model outputs a fatigue level ranging from 0 to 3.

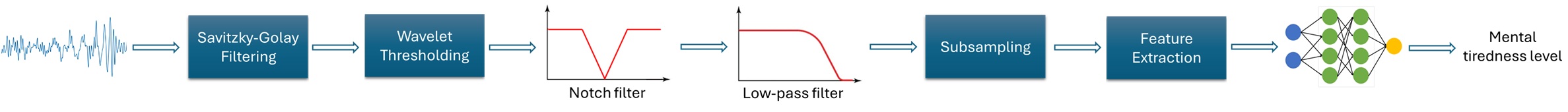


Figure 9: Entire process pipeline.

The classification algorithm presented in Figure 10 makes one prediction every 32 seconds. After accumulating 18 predictions (with 5 needed for the initial notification), the **statistical mode** of the buffer is computed to determine the final predicted fatigue level. If this level is greater than or equal to a predefined threshold (value 2), a notification is sent suggesting that the user take a break. Conversely, if the predicted level is below the threshold, a notification is still sent encouraging the user to resume their study session. In either case, once a notification is sent, the entire prediction buffer is cleared. If no notification is triggered, only the oldest prediction is removed from the buffer.

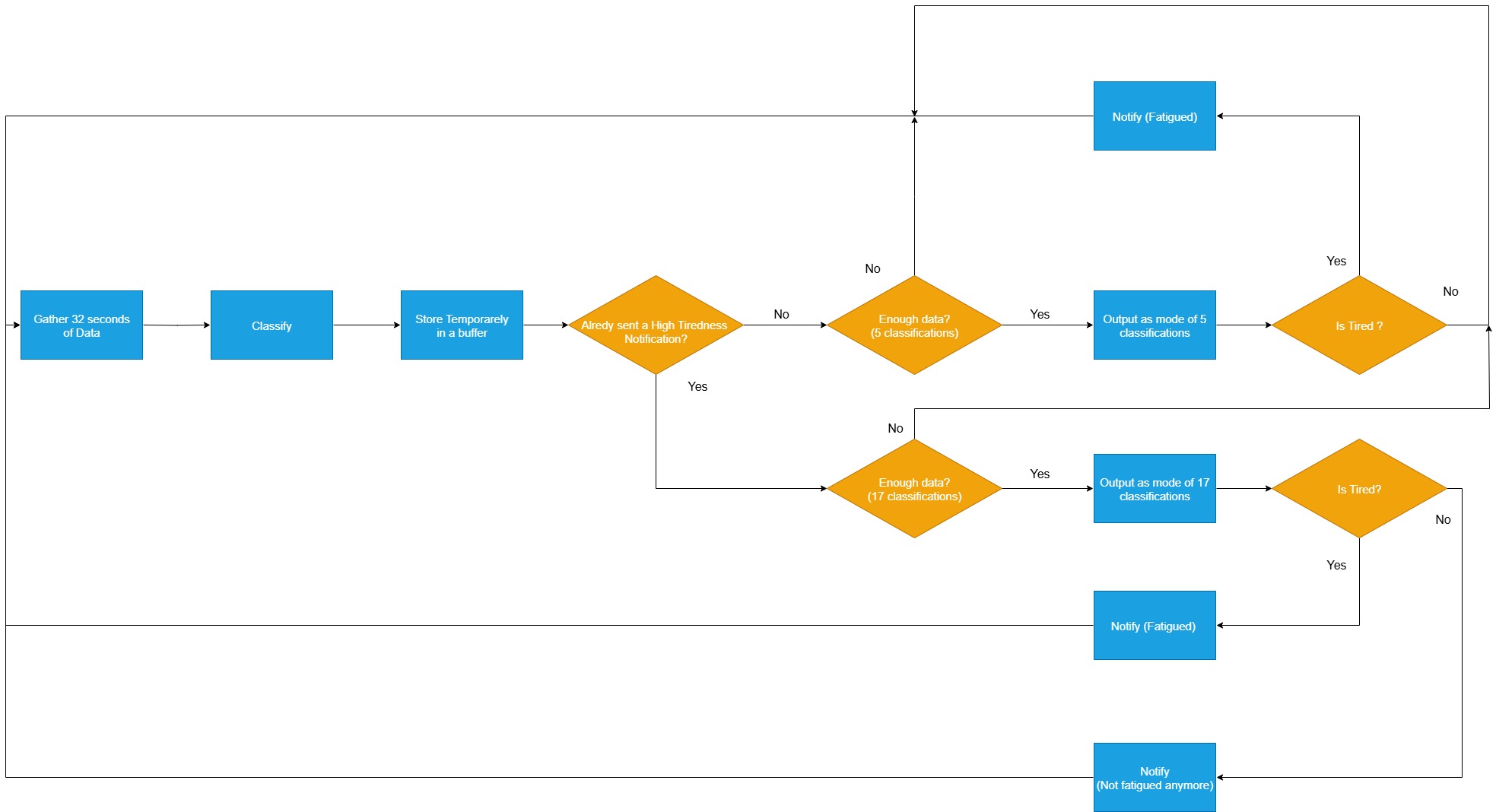


Figure 10: Classification algorithm.

3.1.2 Fine Tuning Service

The second service, called FineTuningService, is explicitly initiated by the user. This service performs **incremental training** (fine-tuning) of the baseline neural network using locally collected EEG data, with the goal of improving the model’s predictive accuracy for the specific user. Once activated, the training process is handled in the background by a separate thread.

If a previously personalized model is available, it is loaded by the service; otherwise, the baseline model is used (all models are in **.tflite** format).

The system then retrieves EEG samples from each study session stored in the local database and checks whether there are enough sessions available for effective fine-tuning. If not, the service will terminate. During training, the sessions are shuffled at the start of each epoch to improve the model’s generalization. Each session contains **16,000 labeled EEG samples**, from which features are extracted using the EegFeatureExtractor module, which also performs **subsampling**.

The resulting feature vectors, along with their corresponding fatigue labels, are used to train the model via the embedded training signature. The training is run for twenty epochs to reinforce learning, this number was choosen to pay attention to the battery consumption and the computational load taking about 15 seconds for the fine tuning to complete on the tested device.

At the end of the training phase, the updated model weights are saved locally using the save signature, which generates a checkpoint file, making the model available for future incremental training sessions. Finally, the service deletes the sessions used from the local database, releases any allocated resources, and terminates successfully.

4 Training of the baseline model

The base model, is a simple Feed forward neural network, this choice is the result of different tries done with other models (Xgboost, Decision tree, Random forest), they achieved comparable results, so the FNN resulted to be the best choice for the compatibility with TensorFlowLite.

4.1 The Choice of TensorFlow Lite

The application uses a classification model in TensorFlow Lite (.tflite) format to analyze EEG data collected from the user and estimate their level of mental fatigue.

TensorFlow Lite was chosen for its ability to efficiently run machine learning models directly on mobile devices. It is designed to be lightweight and optimized, allowing for low inference times, reduced resource consumption (CPU, memory, battery), and compatibility with Android. Thanks to TensorFlow Lite, it is possible to integrate pre-trained neural models into the app and perform real-time inference, even on devices with limited hardware capabilities, without the need for a connection to an external server [8].

Training a model from scratch directly on the user's mobile device was considered too resource-intensive in terms of computation and execution time. Therefore, a base model was pre-trained externally and then integrated into the app, where it can later be personalized through fine-tuning.

4.2 Model Architecture

The used model is a feed-forward neural network, one of the simplest and most widely used architectures in deep learning. It consists of a sequence of layers where information flows in a single direction from the input layer, through one or more hidden layers, to the output layer without cycles or recurrent connections.

Each neuron in a layer is fully connected to the neurons in the next layer and applies a linear transformation followed by a ReLU non-linear activation function. This type of architecture is particularly suitable for classification tasks, where the goal is to assign input to one of several predefined categories.

In our implementation, the feed-forward network is structured as follows:

* **Head** (input layer): 90 neurons, ReLU activation
* **HL1** (hidden layer 1): 64 neurons, ReLU activation
* **HL2** (hidden layer 2): 32 neurons, ReLU activation
* **HL3** (hidden layer 3): 16 neurons, ReLU activation
* **Tail** (Output layer): 4 neurons (corresponding to the 4 fatigue classes), SoftMax activation

The first three layers are frozen during fine-tuning to preserve the knowledge acquired during pre-training and to reduce the computational load. The model also includes some custom methods called “Signatures” which are necessary to allow the personalization of the model on the device [8]. These signatures are:

* **Train**, allows to train the model with training data;
* **Classify**, used for classify an input;
* **Save**, for saving the trainable weights, in a chekpoint file;
* **Load\_weights**, to load the weights from a checkpoint file.

The signatures allow to execute those operations after the conversion to .tflite, making it possible to perform on-device training.

4.3 Training Dataset

The baseline model was pre-trained on a public dataset comprising EEG recordings from 40 subjects to monitor induced stress [10].  
This dataset contains 32-channel EEG 100 Hz recordings of 40 subjects performing four types of cognitive tasks:

* Mental arithmetic
* Stroop color-word test
* Symmetrical image recognition
* Resting period

Each recording is associated with a mental fatigue label ranging from 0 to 10, is important to mention that the distribution of labels is unbalanced. Data are provided as matrices representing EEG activity during each task performed by a given subject.

4.4 Label Reorganization

Since the final model performs classification into 4 classes, whereas the original dataset contains 11 fatigue levels (0 to 10), the labels were grouped as follows:

* **Class 0** (Rest): levels {0, 1}
* **Class 1** (Not fatigued): levels {2, 3, 4}
* **Class 2** (Slightly fatigued): levels {5, 6}
* **Class 3** (Fatigued**)**: levels {7, 8, 9, 10}

This mapping was chosen to ensure a numerically balanced distribution of examples across the four classes.

4.5 Preprocessing and Feature Extraction

In this project, a preprocessing pipeline was introduced to enhance the quality of EEG signals prior to feature extraction. The primary motivation behind this step is to reduce the influence of low-frequency drifts and high-frequency noise, which are common in raw EEG recordings and can negatively impact the accuracy and robustness of downstream machine learning models.

These preprocessing operations are the same as those applied to the pre-training dataset [11]

The preprocessing is composed of two main stages:

* **Trend Removal using Savitzky-Golay Filtering** [12]  
  To eliminate slow-varying baseline fluctuations often caused by sensor drift or subject movement, a Savitzky-Golay filter is applied to each EEG channel. This filter was implemented in Kotlin using fixed convolution coefficients generated by MATLAB's sgolayfilt function with parameters order=5 and framelen=127. The result is a smoothed version of the signal representing the trend, which is then subtracted from the original data to obtain a detrended signal.
* **Noise Reduction using Wavelet Thresholding** [13]  
  After detrending, the signal is denoised using discrete wavelet transform with the Daubechies2 wavelet. The signal is decomposed into five levels of coefficients: one approximation and four detail components. Soft thresholding is applied to each component individually, using a dynamic threshold based on the standard deviation of the third detail level (cd3). This approach suppresses high-frequency components associated with noise while preserving important signal features. Finally, the cleaned signal is reconstructed using the inverse wavelet transform.
* **Notch filtering**, at 50 Hz to reduce power line noise.

This preprocessing pipeline ensures that the extracted features are based on a more stable and noise-free representation of the EEG signal, thereby improving the reliability of the subsequent analysis and classification tasks.

4.6 Feature Extraction

Only the central EEG channels (C) were retained, as they are compatible with the MindRove EEG device used in the app. From the raw EEG voltage signals, 29 features were extracted, including both time-domain and frequency-domain metrics.

Time-domain features:

* MIN, MAX, MEAN
* RMS (Root Mean Square)
* VAR (Variance)
* STD (Standard Deviation)
* POWER (Sum of squared values)
* PEAK, P2P (Peak-to-peak amplitude)
* CREST FACTOR: Ratio of peak value to RMS
* SKEW
* KURTOSIS
* FORM FACTOR: Ratio of RMS to mean
* PULSE INDICATOR: Presence or strength of pulsed activity

Frequency-domain features:

* Absolute and relative power for each EEG band:
  + Delta, Theta, Alpha, Beta, Gamma
* Spectral ratios: [14]
  + (Alpha + Theta) / Beta
  + Alpha / Theta
  + Theta / Alpha
* Peak frequency
* Spectral entropy

Each time window is represented as a 6×29 matrix (29 features for each of the 6 EEG channels).

4.7 Feature Selection

During training, a feature selection process was applied to identify the most relevant features for model performance [1].

The most relevant input features for classification were identified through analysis of the weights of the neural network. In the first dense layer, each input sample is represented as a flattened vector of shape 6×29=174, where 6 refers to EEG channels and 29 to extracted features per channel. The first dense layer processes this 174-dimensional input and maps it onto 64 neurons, resulting in a weight matrix of shape 174×64.

The absolute values of the weights associated with each input element (i.e., each channel-feature pair) were summed across all neurons. To estimate the global importance of each feature, importance scores were aggregated across all channels. The 15 features with the highest cumulative weight magnitudes were selected as the most significant. From the original 29, the following 15 features were selected:

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

The final base model was trained on 6×15 matrices, containing the above 15 features extracted from each of the six selected EEG channels.

4.8 Preliminary Results

In order to asses the performances of the pre-trained model, a 10-fold stratified cross validation was performed.

To establish a performance baseline, the pre-trained model was evaluated using 10-fold cross-validation on the test set of the original dataset. The aggregated results, which will be compared against our personalized model in Section 5, show a macro-average F1-score of 0.25 and an overall accuracy of 25%. Achieving results of a random chance classifier, this performance is typical for a generic model on noisy, subject-independent EEG data and the limited amount of training data (Initial small dataset, then subsampled to make it balanced). The model serves as the starting point for on-device personalization.

It is important to highlight that this is only an initial model, not yet personalized to the user’s real data. Its primary purpose is to provide a solid foundation for further adaptation via fine-tuning in real-world usage.

5 Experimental results

5.1 Experimental Setup

To validate the on-device personalization pipeline, an extensive data collection and evaluation was performed. A single user (one of the project authors) used the Selene application with a MindRove Arc headset and a Redmi 13C smartphone (Android 14) to record 34 distinct study sessions in a quiet indoor environment. After each session, a self-reported fatigue label (0-3) was provided, operationalizing the humans-as-sensors paradigm. This dataset, stored locally in the app's Room database, formed the basis for fine-tuning and evaluation using a 10-fold cross-validation methodology.

5.2 Results and Discussion

To quantitatively evaluate the effectiveness of the on-device personalization, we compared the performance of the generic baseline model against the fine-tuned model. The evaluation was conducted using a 10-fold cross-validation methodology. The aggregated results are presented below.

The performance of the baseline model, as shown in Table 1, was modest. With an overall accuracy of 25%, it performed only slightly better than random chance (25% for a 4-class problem). The precision, recall, and F1-scores for all classes are low, hovering between 0.20 and 0.31, is consistent with expectations for subject-independent EEG classification. Such models often struggle to generalize across individuals due to high inter-subject neurophysiological variability, a well-documented challenge in the field [2], [3], [4]. This baseline performance, equivalent to random chance for a four-class problem, serves as a crucial control, demonstrating that a generic model is insufficient for this nuanced task. The confusion matrix in Figure 11 further illustrates this, showing that predictions are scattered across all classes with no clear diagonal pattern. This indicates that the baseline model struggled to distinguish between the different levels of fatigue.

Table 1: Performance metrics of the baseline model aggregated over 10 folds.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 |

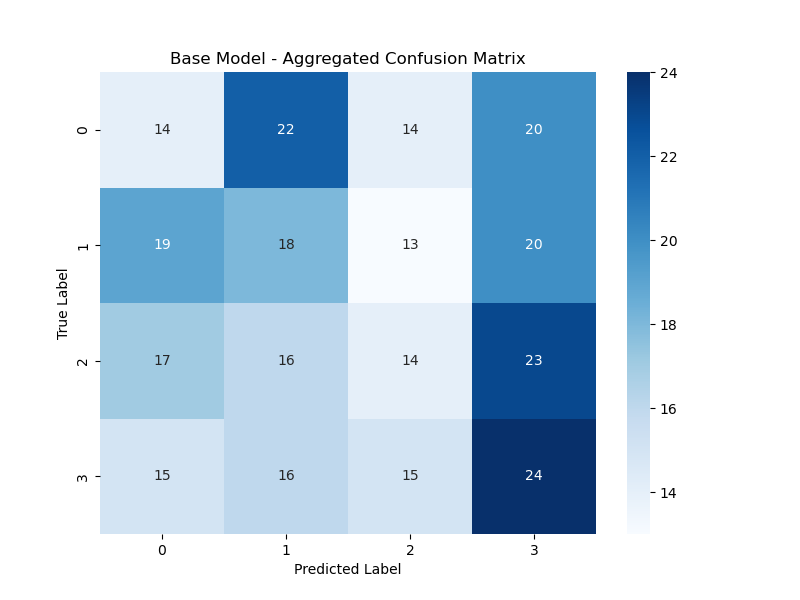


Figure 11: Aggregated confusion matrix for the baseline model.

In stark contrast, the personalized model showed a dramatic improvement in performance after being fine-tuned on the user's data. As detailed in Table 2, the overall accuracy jumped from 25% to 55%.

Table 2: Performance metrics of the personalized (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 |

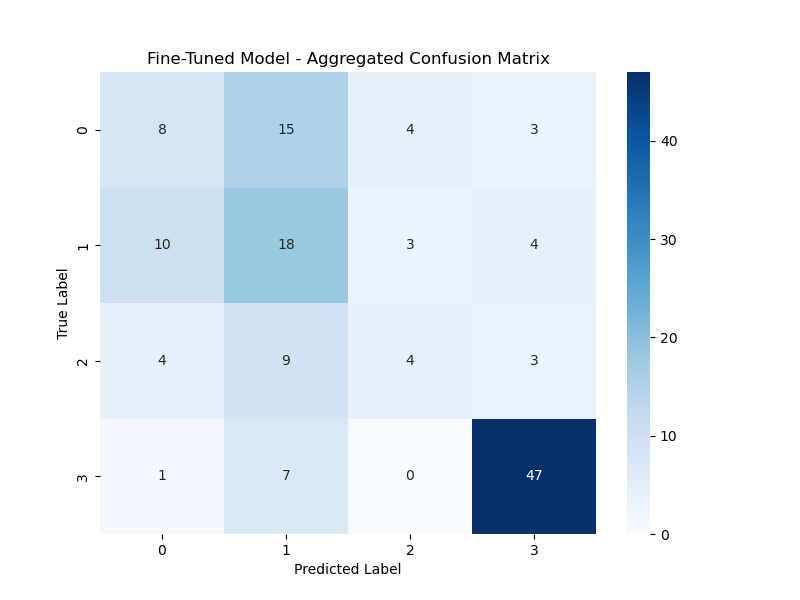


Figure 12: Aggregated confusion matrix for the personalized model.

The most significant improvement is observed for Class 3 (maximum fatigue), which achieved an F1-score of 0.84, with a precision of 0.82 and a recall of 0.85. The confusion matrix in Figure 12 clearly visualizes this success: out of 55 instances of maximum fatigue, the model correctly identified 47. This result strongly supports the efficacy of the humans-as-sensors paradigm, where incorporating even a limited amount of subjective, user-specific data significantly enhances model performance [2]. The ability of the fine-tuned model to reliably identify high-fatigue states is particularly significant, as this is the most critical state for triggering practical interventions like break suggestions

. While the model still shows some confusion between classes 0, 1, and 2, the performance for Class 1 (F1-score of 0.43) is also markedly better than the baseline. This demonstrates that even a small amount of personal data significantly enhances the model's ability to recognize user-specific EEG patterns.

Further analysis reveals the model's learned behavior. Table 3 shows the average predicted class for each true class. For the personalized model, when the true class was 3, the average prediction was **2.69**, very close to the actual label. This indicates the model is not just guessing but has learned a strong signal for high fatigue. Conversely, the confusion between classes 0, 1, and 2 is reflected in their average predictions (1.07, 1.03, and 1.30), suggesting the EEG patterns for low-to-moderate fatigue are more subtle or overlapping for this user. It also shows that imbalanced dataset collected by the user can influence the prediction of the final mode. With the highest number of sessions taken at 1 and 3 classes the model has a decent discriminative power between the low and maximum fatigue levels. However it also tends to bias especially towards the low mental workload level when no apparent deviations from “no fatigue” and “some fatigue” are present.

Table 3: The average predicted fatigue level for each actual fatigue class.

|  |  |
| --- | --- |
| True Class | Average Predicted Class |
| 0 | 1.07 |
| 1 | 1.03 |
| 2 | 1.30 |
| 3 | 2.69 |

6 Conclusion

This project, Selene, successfully demonstrates the feasibility of a complete, on-device system for personalized mental fatigue monitoring. By integrating an external EEG sensor with a smartphone application, we have created a pipeline that handles real-time data acquisition, local storage, on-device machine learning inference, and user-driven model personalization. Our work makes a practical contribution to the field of mobile and social sensing systems by showcasing how a smartphone can act as a central hub for personal wellness applications [6], [8].

The main finding of our experimental evaluation is the significant and tangible benefit of personalization, which improved the model's overall accuracy from a baseline of 25% to 55% and, crucially, increased the F1-score for detecting maximum fatigue from 0.31 to 0.84. The fine-tuning process, which leverages the humans-as-sensors paradigm by incorporating subjective user feedback, demonstrably improved the accuracy and relevance of fatigue predictions. This confirms our initial hypothesis that a one-size-fits-all model is insufficient for a nuanced and personal state like mental fatigue [2], [4], and that on-device learning is a powerful tool for creating truly user-centric applications. The architecture proved capable of supporting continuous, real-time monitoring, although the performance is limited and the model works best with a considerable amount of datapoints i.e. labeled session of study.

Despite the promising results, this work has several limitations. The evaluation was conducted on a single subject (n=1), so the results are not generalizable. The ground truth itself, based on self-reporting, is inherently subjective [15]. Furthermore, the system is dependent on the MindRove Arc hardware and its power-intensive Wi-Fi connection.

Future work could test Selene with a larger and more diverse group of users to validate its effectiveness more broadly. The system could be enhanced by incorporating data from other built-in smartphone sensors, such as the accelerometer, to add contextual awareness (e.g., detecting if the user is stationary), a common strategy in multimodal sensing systems [3], [8] or correlating study times with the time of day could provide richer contextual cues to the model. Further work could also explore switching to more precise EEG devices utilizing more electrodes at measuring signals from different brain areas. Devices allowing to change from Wi-Fi to Bluetooth Low Energy (BLE) for communication with the headset, if supported, could significantly improve the energy efficiency of the system and reduce battery drain on the mobile device [6], [8].

In conclusion, Selene represents a successful synthesis of mobile application development, wearable sensing, and on-device machine learning. It serves as a strong proof-of-concept for how modern mobile systems can be engineered to provide personalized, context-aware insights that empower users to better understand and manage their cognitive well-being.

REFERENCES

[1] IBM, “What is feature selection?” Accessed: Jul. 12, 2025. [Online]. Available: https://www.ibm.com/think/topics/feature-selection

[2] J. Yauri, A. Hernández-Sabaté, P. Folch, and D. Gil, “Mental Workload Detection Based on EEG Analysis,” in *Frontiers in Artificial Intelligence and Applications*, IOS Press BV, Oct. 2021, pp. 268–277. doi: 10.3233/FAIA210144.

[3] Y. Zhou, S. Huang, Z. Xu, P. Wang, X. Wu, and D. Zhang, “Cognitive Workload Recognition Using EEG Signals and Machine Learning: A Review,” Sep. 01, 2022, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/TCDS.2021.3090217.

[4] K. Kingphai and Y. Moshfeghi, “Mental Workload Assessment Using Deep Learning Models From EEG Signals: A Systematic Review,” 2025, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/TCDS.2024.3460750.

[5] S. Shao, G. Han, T. Wang, C. Lin, C. Song, and C. Yao, “EEG-Based Mental Workload Classification Method Based on Hybrid Deep Learning Model Under IoT,” *IEEE J Biomed Health Inform*, vol. 28, no. 5, pp. 2536–2546, May 2024, doi: 10.1109/JBHI.2023.3281793.

[6] R. Zanetti, A. Arza, A. Aminifar, and D. Atienza, “Real-Time EEG-Based Cognitive Workload Monitoring on Wearable Devices,” *IEEE Trans Biomed Eng*, vol. 69, no. 1, pp. 265–277, Jan. 2022, doi: 10.1109/TBME.2021.3092206.

[7] D. Antolinez *et al.*, “Development and Validation of an Affordable Wearable Multichannel EEG System With Active Electrodes and Innovative Flexible Headbands,” *IEEE Access*, vol. 13, pp. 83983–83995, 2025, doi: 10.1109/ACCESS.2025.3568302.

[8] TensorFlow Authors, “On-device training with TensorFlow Lite”, Accessed: Jul. 12, 2025. [Online]. Available: https://www.tensorflow.org/lite/training/on\_device\_training

[9] MindRove, “MindRove Arc.” Accessed: Jul. 12, 2025. [Online]. Available: https://mindrove.com/arc/

[10] A. Appriou, E. Cvetkovic, T. L. T. T. Nguyen, P. V. F. B. Dos Santos, and F. B. Vialatte, “SAM40 Dataset,” 2021, *Figshare*. Accessed: Jul. 12, 2025. [Online]. Available: https://figshare.com/articles/dataset/SAM40\_Dataset/14349359

[11] A. Appriou, E. Cvetkovic, T. L. T. T. Nguyen, P. V. F. B. Dos Santos, and F. B. Vialatte, “Original MATLAB Preprocessing Script,” 2021, *Figshare*. Accessed: Jul. 12, 2025. [Online]. Available: https://figshare.com/articles/software/Original\_MATLAB\_Preprocessing\_Script/14349380

[12] Inc. The MathWorks, “Savitzky-Golay filtering.” Accessed: Jul. 12, 2025. [Online]. Available: https://www.mathworks.com/help/signal/ref/sgolayfilt.html

[13] Inc. The MathWorks, “Wavelet Transforms: A Practical Guide.” Accessed: Jul. 12, 2025. [Online]. Available: https://www.mathworks.com/discovery/wavelet-transforms.html

[14] B. Raufi and L. Longo, “An Evaluation of the EEG Alpha-to-Theta and Theta-to-Alpha Band Ratios as Indexes of Mental Workload,” *Front Neuroinform*, vol. 16, May 2022, doi: 10.3389/fninf.2022.861967.

[15] A. Hamann and N. Carstengerdes, “Don’t Think Twice, It’s All Right? – An Examination of Commonly Used EEG Indices and Their Sensitivity to Mental Workload,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, 2023, pp. 65–78. doi: 10.1007/978-3-031-35392-5\_5.