



Algorithm for Brain-based Authentication System

1. Input:

- EEG/EOG/SEMG dataset (eeg_eog_semg_dataset_large.csv).
- Features: Signal data (Signal1, Signal2, Signal3).
- Labels: Target variable (Label).

2. Preprocessing:

- Load the dataset into a DataFrame.
- Encode categorical labels into numerical format using **Label Encoder**.
- Standardize the signal data using **Standard Scaler** to normalize feature values.

3. Data Splitting:

- Divide the dataset into training and testing subsets using an 80-20 split with **train_test_split**.
- Inputs (X_train, X_test) are the standardized signals.
- Labels (y_train, y_test) are the encoded target values.

4. Model Training:

- Initialize a **Random Forest Classifier** with 100 estimators and a fixed random state for reproducibility.
- Train the model on the training dataset (X_train, y_train).

5. Prediction and Evaluation:

- Predict labels for the test data (y_test) using the trained model.
- Calculate the **accuracy score** of the predictions.
- Generate a **classification report** to evaluate model performance, ensuring proper mapping of labels to class names.

6. Result Reporting:

For each test sample:

- Compare the predicted label with the true label.
- If labels match: Print "Access granted."
- If labels do not match: Print "Access denied."



7. Visual Feedback:

- Plot a subset of test signals (e.g., the first 50 samples) to visualize the standardized signal values.

8. Output:

- Display the model accuracy.
- Present the detailed classification report with precision, recall, F1-score, and support for each class.
- Show results for each test sample (match/mismatch).
- Display the visualization of test signals.

Steps Summary (Pseudocode):

1. Load dataset and initialize tools.
2. Encode labels numerically.
3. Standardize signal features.
4. Split data into training and testing sets.
5. Train a Random Forest Classifier on the training set.
6. Predict labels for the testing set.
7. Evaluate and display:
 - Accuracy of the model.
 - Classification report with metrics.
8. For each test sample:
 - Compare predicted vs. true label.
 - Print "Access granted" or "Access denied."
9. Optionally plot a subset of test signals for visualization.



Flow Chart:

