Assessing Neurological States from Physiological Signals

The American University in Cairo
Sama Amr Gouda – 900211296
Mona Ibrahim – 900212749
Shaymaa El Sayed – 900214260

Author Note

This research paper is written for the DSCI 4411 course, under the supervision of Dr. Seif Eldawlatly, Department of Mathematics, Actuarial Science and Data Science at the American University in Cairo.

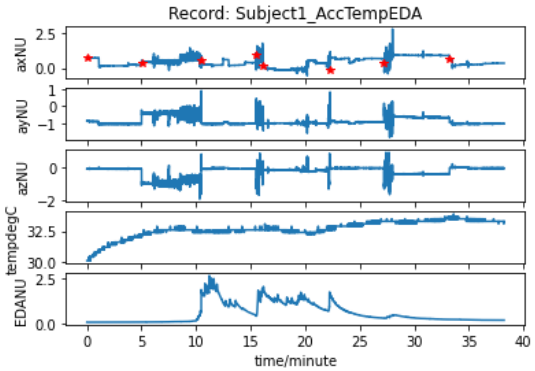
Table of Content

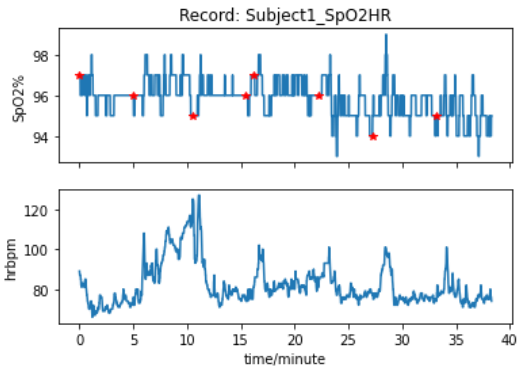
Introduction

Neurological health is a critical aspect of overall well-being, with conditions affecting the nervous system often having profound implications on quality of life. Traditionally, an EEG (electroencephalography) is used to measure the electrical activity in the brain, helping in diagnosing the brain condition (*EEG (Electroencephalogram)*, 2024). However, it's resource intensive and requires specialized equipment. Thus, Non-EEG physiological signals examination has been introduced where it "evaluates brain and nervous system functioning" (Cleveland Clinic Medical, 2022). The exam is basically a physical examination to identify the signs of disorders affecting the brain (Cleveland Clinic Medical, 2022). This report is going to address the Non-EEG Dataset for the Assessment of Neurological Status, a database collected at the Quality of Life Laboratory at the University of Texas at Dallas, in various different ways with the potential to classify these collected physiological signals into either one of the four neurological states: physical stress, cognitive stress, emotional stress and relaxation. Thus, this is defined to be a classification problem with 4 classes.

Dataset Description

Our dataset is retrieved from Physionet and is collected at the Quality of Life Laboratory at University of Texas at Dallas.The data has been collected using non-invasive wrist worn biosensors, which measure 5 different attributes: electrodermal activity (EDA), temperature, acceleration, heart rate (HR), and arterial oxygen level (SpO2). This was gathered for a total of 20 subjects, where each had to undergo a series of 7 stages of which include: 4 Relaxation periods spread in between all the states, physical stress, an emotional stress period, and a combination of a cognitive stress and a mini- emotional stress period. EDA, temperature, and acceleration were all recorded using 8 ticks per second- 4 for relaxation, 2 for emotional stress, 1 for physical stress and 1 for cognitive stress- over a 35 minute period. As for HR, and SpO2 they were recorded using 1 tick per second over a 35 minute period as well. Thus, the data is compiled into 2 files for each of the subjects, 1 accounting for all three attributes EDA, temperature, and acceleration, and another for both HR, and SpO2. Finally, an excel file is collected containing info about each of its subjects with their age, height, weight, and gender.





Data preprocessing

1. Initial Signal Preprocessing

The preprocessing involves preparing physiological signals for analysis by aligning, segmenting, and organizing data from two types of files:

a. **Signal Types and Sampling Rates**

The AccTempEDA file contains signals for accelerometer axes (ax, ay, az), temperature, and electrodermal activity (EDA), sampled at 8 Hz (8 samples per second).The SpO2HR file includes signals for oxygen saturation (SpO2) and heart rate (HR), sampled at 1 Hz (1 sample per second).

b. Annotation Parsing

From the annotation file, we extract: Sample indices (sample): These define the start and end points of each labeled stage. The indices correspond to the 8 Hz sampling rate.Stage names (aux_note): These provide the activity or stress stage labels, such as 'PhysicalStress', 'Relax', 'EmotionalStress', and 'CognitiveStress'.

c. Aligning Signals with Different Sampling Rates Since the SpO2HR signals are sampled at 1 Hz, the 8 Hz sample indices are converted to the 1 Hz timeline by dividing them by 8. This ensures that all signals are properly aligned for processing.

d. Signal Segmentation by Stages
Signals are segmented based on the annotation start and stop indices: for each stage (except the last one), data is extracted from sample[j] to sample[j+1].
For the last stage, data is extracted from sample[j] to the end of the recording.
These segmented signals are stored in dictionaries grouped by stage (e.g., 'Relax', 'PhysicalStress', etc.) for each subject.

2. Signal Resizing

To ensure signals are uniform across all subjects and stages, we resize the signals to have a consistent length. This step is important for creating a standardized dataset suitable for further processing or machine learning. For this purpose, we use the variables

minimum_values_accEDA and minimum_values_sp02hr, which represent the minimum lengths for each stage and signal type.

For each subject, stage, and repetition:

a. **Signals Resized Based on Type:**

For accelerometer axes (ax, ay, az), EDA, and temperature signals, the minimum lengths are taken from minimum_values_accEDA (e.g., 2384 samples for 'Relax', 2593 for 'PhysicalStress', etc.). For SpO2 and heart rate (spo2, hr), the minimum lengths are taken from minimum_values_sp02hr (e.g., 298 samples for most stages and 40 samples for 'EmotionalStress').

b. **Resizing**:

Signals shorter than the minimum length are padded or extended.Signals longer than the minimum length are truncated.This resizing ensures uniformity across stages and subjects, making all signal segments comparable and ready for concatenation into a unified feature vector.

c. **Repetition of stages** :

Since some stages repeat multiple times (e.g., 'Relax' repeats 4 times), this process is applied separately to each repetition, ensuring every instance of a stage is resized appropriately.

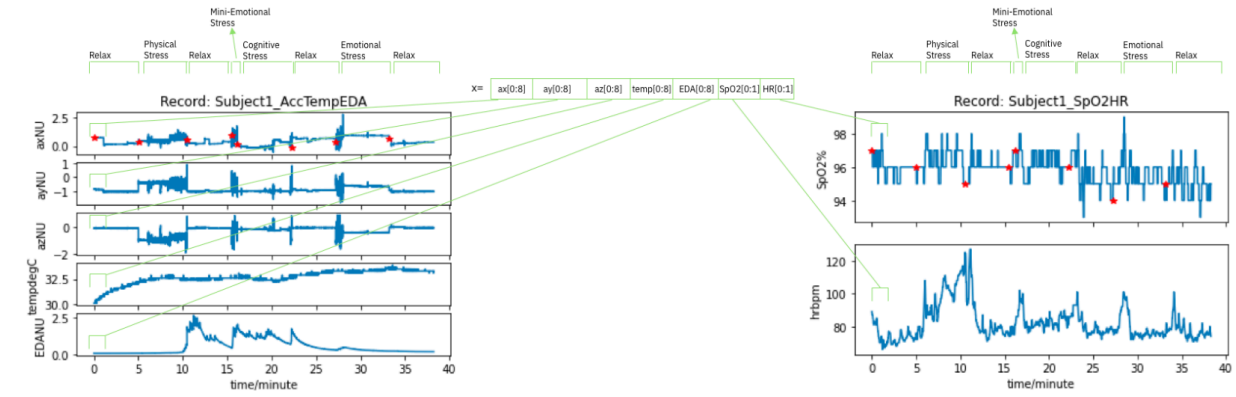
3. **Feature and Label Extraction**

In this part we are splitting the signal into multiple observations for each subject. Each observation is a 42-dimensional feature vector, formed by concatenating the values from the AccTempEDA signals and the SpO2HR signals.

- 40 values come from the AccTempEDA signals (as we have five signals (e.g., ax, ay, az, temp, EDA) contributes 8 values to the observation)
- 2 additional values come from the SpO2HR signals (1 sample from SpO2 and 1 sample from HR).

Thus, each observation vector has a total of 40 (from ‘AccTempEDA‘)+1 (from SpO2)+1 (from HR)=42 . Afterwards, the signal is divided into segments based on the sample indices derived from annotations, typically with a fixed length.

For each segment, the corresponding samples from each signal (ax, ay, az, temp, EDA, SpO2, and HR) are concatenated to form a feature vector.



Approach for modelling

Once the dataset is prepared and structured, we can proceed with the modeling phase. The goal here is to classify the four stages of activity (Relax, PhysicalStress, EmotionalStress, CognitiveStress) for each subject. We will train several machine learning models separately for each subject.

Results

We have explored various machine-learning models to classify neurological states using physiological signals. The performance of each model was evaluated when using cross-validation with PCA, leave-one-out with PCA, and cross-validation without PCA.

Multi-Layer Perceptron:

	Model	Cross-Validation	Leave-one out	PCA	Accuracy
0	Multi-Layer Perceptron	Yes	---	Yes	98.850267
1	Multi-Layer Perceptron	--	Yes	Yes	66.423983
2	Multi-Layer Perceptron	Yes	--	NO	66.423983

The Multi-Layer Perceptron using cross-validation with PCA achieved an accuracy of **98.85%**, making it the second most accurate model in our project. The Leave-one-out method with PCA significantly reduced the accuracy to **66.42%**, highlighting the limitations of this validation method for the MLP model. MLP excels in high-dimensional, complex data environments. However, it is sensitive to the size and quality of training data. When fewer data points are available (as in leave-one-out), its performance suffers significantly. The MLP using cross validation without PCA had a similar performance as the leave-one out method, showing that using PCA with this method enhanced it.

K-Nearest Neighbors (KNN):

	Model	Cross-Validation	Leave-one out	PCA	Accuracy
0	Multi-Layer Perceptron	Yes	---	Yes	98.850267
1	Multi-Layer Perceptron	--	Yes	Yes	66.423983
2	Multi-Layer Perceptron	Yes	--	NO	66.423983
3	KNN	Yes	---	Yes	93.676471
4	KNN	---	Yes	Yes	71.616702
5	KNN	Yes	--	NO	91.483957

KNN with cross-validation and PCA achieved **93.68% accuracy** showing that it performed well to classify neurological states. The Euclidean distance metric used by KNN worked well in the transformed feature space. Having cross validation without PCA decreased the model accuracy to **91.48%** indicating that using the PCA enhanced the model performance. Lastly, the Leave-one-out validation with PCA yielded **71.62% accuracy**, showing that KNN struggled to maintain its robustness with reduced data in each fold.

Random Forest:

	Model	Cross-Validation	Leave-one out	PCA	Accuracy
0	Multi-Layer Perceptron	Yes	---	Yes	98.850267
1	Multi-Layer Perceptron	--	Yes	Yes	66.423983
2	Multi-Layer Perceptron	Yes	--	NO	66.423983
3	KNN	Yes	---	Yes	93.676471
4	KNN	---	Yes	Yes	71.616702
5	KNN	Yes	--	NO	91.483957
6	Random Forest	Yes	---	Yes	98.796791
7	Random Forest	---	Yes	Yes	77.550857
8	Random Forest	Yes	---	NO	99.184492

Random Forest offers high accuracy and robustness against overfitting due to its ensemble approach. However, it was a bit computationally intensive, as it took longer to train than other models. It had **98.80% accuracy** under cross-validation with PCA highlighting its strength in capturing non-linear interactions and handling noise effectively. Its ensemble nature enables it to generalize well across diverse feature spaces. The Random Forest performed the best compared to all other methods used in our project, when using cross validation without PCA. It performed with an accuracy of **99.18%**. However, again with leave-one-out validation, accuracy dropped to **77.55%**, reflecting the challenges of training each decision tree with limited data, and consequently affecting the accuracy.

Since the Random Forest yielded the best results among the other models, we decided to make use of Random Forest to get the feature importance; therefore, we used the “feature_importances_” command to extract it from our model. This works by utilizing the model's internal calculations to measure feature importance, by using the Gini importance, which is how the feature brings about total reduction to the criterion, which is the Gini impurity. Basically, this gives more weight to the most important features, that leads to the greater impurity reduction, by measuring the impurity of a node in a decision tree. It extends to evaluate the contribution of each feature across multiple trees. On the right is the output of this execution showing the Gini importance of each of the features, with the maximum importance being assigned to feature 17 that is related to “az” which is part of “EDA.”

```
Max importance feature:  17
Importance:  0.0955849550981549
Value Related to az
```

Feature	Importance
1	0.003918
2	0.008186
3	0.004017
4	0.005090
5	0.005818
6	0.004335
7	0.005105
8	0.001730
9	0.003299
10	0.001519
11	0.005849
12	0.002760
13	0.004015
14	0.001629
15	0.003640
16	0.003829
17	0.095585
18	0.088780
19	0.057243
20	0.046901
21	0.061616
22	0.057379
23	0.070388
24	0.048101
25	0.027703
26	0.037728
27	0.019445
28	0.035874
29	0.033395
30	0.024860
31	0.045023
32	0.020402
33	0.011443
34	0.011662
35	0.012234
36	0.008507
37	0.010997
38	0.009636
39	0.017331
40	0.011866
41	0.018856
42	0.052308

Decision Tree:

	Model	Cross-Validation	Leave-one out	PCA	Accuracy
0	Multi-Layer Perceptron	Yes	---	Yes	98.850267
1	Multi-Layer Perceptron	--	Yes	Yes	66.423983
2	Multi-Layer Perceptron	Yes	--	NO	66.423983
3	KNN	Yes	---	Yes	93.676471
4	KNN	---	Yes	Yes	71.616702
5	KNN	Yes	--	NO	91.483957
6	Random Forest	Yes	---	Yes	98.796791
7	Random Forest	---	Yes	Yes	77.550857
8	Random Forest	Yes	---	NO	99.184492
9	Decision Tree	Yes	---	Yes	93.930481
10	Decision Tree	---	Yes	Yes	77.186831
11	Decision Tree	Yes	---	NO	95.641711

Decision Trees are simple and interpretable, performing well in datasets with multiple features. However, they can be prone to overfitting without regularization or ensemble techniques like Random Forest. In case of our dataset, the decision trees performed a bit lower than the Random Forest. It had an accuracy of **93.93%** with cross-validation and PCA. A higher accuracy of **95.64%** was observed when PCA was excluded, and finally, the Leave-one-out validation with PCA resulted in **77.19% accuracy**, which is the lowest of the three methods.

Naive Bayes:

	Model	Cross-Validation	Leave-one out	PCA	Accuracy
0	Multi-Layer Perceptron	Yes	---	Yes	98.850267
1	Multi-Layer Perceptron	--	Yes	Yes	66.423983
2	Multi-Layer Perceptron	Yes	--	NO	66.423983
3	KNN	Yes	---	Yes	93.676471
4	KNN	---	Yes	Yes	71.616702
5	KNN	Yes	--	NO	91.483957
6	Random Forest	Yes	---	Yes	98.796791
7	Random Forest	---	Yes	Yes	77.550857
8	Random Forest	Yes	---	NO	99.184492
9	Decision Tree	Yes	---	Yes	93.930481
10	Decision Tree	---	Yes	Yes	77.186831
11	Decision Tree	Yes	---	NO	95.641711
12	Naive Bayes	Yes	---	Yes	90.508021
13	Naive Bayes	---	Yes	Yes	72.526767
14	Naive Bayes	Yes	---	No	91.764706

The Naive Bayes achieved **90.51% accuracy** under cross-validation with PCA. Excluding PCA slightly improved accuracy to **91.76%**, suggesting that Naive Bayes could benefit from the original dataset's feature distribution, and the accuracy dropped to dropped to **72.53%** when using the leave one out method.

Fisher's Linear Discriminant Analysis (FLDA):

	Model	Cross-Validation	Leave-one out	PCA	Accuracy
0	Multi-Layer Perceptron	Yes	---	Yes	98.85
1	Multi-Layer Perceptron	--	Yes	Yes	67.57
2	Multi-Layer Perceptron	Yes	--	NO	67.57
3	KNN	Yes	---	Yes	93.68
4	KNN	---	Yes	Yes	71.62
5	KNN	Yes	--	NO	91.48
6	Random Forest	Yes	---	Yes	98.74
7	Random Forest	---	Yes	Yes	76.51
8	Random Forest	Yes	---	NO	99.12
9	Decision Tree	Yes	---	Yes	93.93
10	Decision Tree	---	Yes	Yes	77.19
11	Decision Tree	Yes	---	NO	95.64
12	Naive Bayes	Yes	---	Yes	90.51
13	Naive Bayes	---	Yes	Yes	72.53
14	Naive Bayes	Yes	---	No	91.76
15	FLDA	Yes	---	Yes	91.86
16	FLDA	---	Yes	Yes	77.24
17	FLDA	Yes	---	No	91.89

FLDA is efficient for linearly separable data and interpretable in terms of feature contributions. However, it struggles with non-linear patterns and small training datasets. The model achieved **91.86% accuracy** with PCA and cross-validation, confirming its capability to find a linear boundary that separates the classes effectively. Without PCA, accuracy slightly improved to **91.89%**, indicating that FLDA can handle raw feature sets (without computing the PCs) efficiently. Finally, the Leave-one-out method with PCA yielded an accuracy of **77.24%** which is the least accuracy among the other methods.

Least squares:

	Model	Cross-Validation	Leave-one out	PCA	Accuracy
0	Multi-Layer Perceptron	Yes	---	Yes	98.85
1	Multi-Layer Perceptron	--	Yes	Yes	67.57
2	Multi-Layer Perceptron	Yes	--	NO	67.57
3	KNN	Yes	---	Yes	93.68
4	KNN	---	Yes	Yes	71.62
5	KNN	Yes	--	NO	91.48
6	Random Forest	Yes	---	Yes	98.74
7	Random Forest	---	Yes	Yes	76.51
8	Random Forest	Yes	---	NO	99.12
9	Decision Tree	Yes	---	Yes	93.93
10	Decision Tree	---	Yes	Yes	77.19
11	Decision Tree	Yes	---	NO	95.64
12	Naive Bayes	Yes	---	Yes	90.51
13	Naive Bayes	---	Yes	Yes	72.53
14	Naive Bayes	Yes	---	No	91.76
15	FLDA	Yes	---	Yes	91.86
16	FLDA	---	Yes	Yes	77.24
17	FLDA	Yes	---	No	91.89
18	LS	Yes	---	Yes	64.33
19	LS	---	Yes	Yes	41.71
20	LS	Yes	---	No	64.28

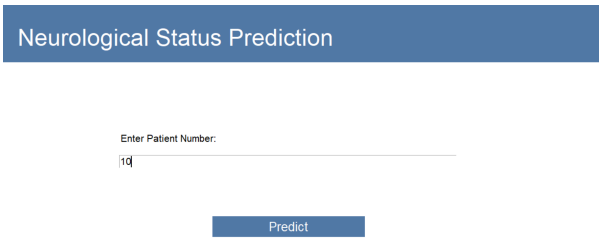
Finally, we tried the Least Squares classification, which is computationally efficient but unsuitable for datasets with non-linear relationships or high feature dimensionality. With a maximum accuracy of only **64.34%** (cross-validation with PCA), LS underperformed significantly. It lacks the flexibility to model complex relationships in the dataset, which explains its poor performance. When trying LS using cross validation without PCA, the accuracy reduced to **64.28%**, showing that it did not enhance the performance. Lastly, the leave-one out method had a very low accuracy of **41.71%**, indicating that it did not work for our model.

Graphical User Interface (GUI)

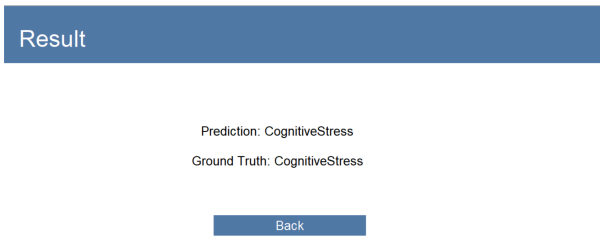
After addressing the different types of classifiers, we found out that the random forest classifier without PCA applied onto the data, achieved the highest accuracy. Thus, we decided to create a GUI that takes the subject number as an input, then it selects a random signal vector at time ‘t’ that contains a combination of the different attributes for that particular subject. It then outputs the classification of the state this vector is at, comparing it to the ground truth. Tkinter library was used to compile this GUI with the help of some basic libraries. As for the running order, we ran it after we trained the Random Forest Classifier, saving it into a pickle file in order for us to be able to access it while running the GUI. This application would better help doctors and Lab-technicians in identifying signals faster, minimizing the inference time where the results are just a click away; in addition to, minimizing the human error. For future work, we would like to

upgrade such an application to take a subject's file as an input even if it isn't contained within the dataset, so that it's more dynamic and can be generalized and used over more numbers of patients.

Input Frame



Output Frame



Conclusion

In this project, we explored the classification of neurological states using physiological signals from the Non-EEG Dataset for the Assessment of Neurological Status. Through detailed data preprocessing and the application of various machine learning models, we demonstrated the feasibility of accurately classifying states such as relaxation, physical stress, emotional stress, and cognitive stress. Our results highlighted the strengths and limitations of different classifiers. Random Forest achieved the highest accuracy, highlighting its robustness in capturing non-linear interactions within the data. Meanwhile, Multi-Layer Perceptron and K-Nearest Neighbors also performed well, demonstrating their capabilities in handling high-dimensional datasets with complex patterns. Conversely, models like Least Squares and Fisher's Linear Discriminant Analysis showed limitations, particularly in non-linear and smaller training datasets. In conclusion, this paper demonstrates the potential of using physiological signals and machine learning for non-invasive neurological assessments, paving the way for more accessible and efficient diagnostic tools in medical and research settings. Future work could focus on expanding the dataset, refining the classifiers, and improving the GUI to further enhance usability and accuracy.

```
Model with the Max Accuracy:  Random Forest
Accuracy:  99.18
```

References

- Cleveland Clinic Medical. (2022, April 1). Neurological exam. Cleveland Clinic. https://my.clevelandclinic.org/health/diagnostics/22664-neurological-exam

- EEG (electroencephalogram). (2024, May 29). Mayo Clinic. https://www.mayoclinic.org/tests-procedures/eeg/about/pac-20393875