Final Report

Project Name:

Unsupervised Learning of EEG States for the Application of Seizure Detection in Epilepsy

Project Number:

2022659

Group members:

Mehak Kalra

Fatima Siddiqui

Khalil Scott

Mina Assaad

ECE496Y

Voluntary Document Release Consent Form

To all ECE496 students:

To better help future students, we would like to provide examples that are drawn from excerpts of past student reports. The examples will be used to illustrate general communication principles as well as how the document guidelines can be applied to a variety of categories of design projects (e.g. electronics, computer, software, networking, research).

Any material chosen for the examples will be altered so that all names are removed. In addition, where possible, much of the technical details will also be removed so that the structure or presentation style are highlighted rather than the original technical content. These examples will be made available to students on the course website, and in general may be accessible by the public. The original reports will <u>not</u> be released but will be accessible only to the course instructors and administrative staff.

Participation is completely voluntary and students may refuse to participate or may withdraw their permission at any time. Reports will only be used with the signed consent of all team members. Participating will have no influence on the grading of your work and there is no penalty for not taking part.

If your group agrees to take part, please have all members sign the bottom of this form.

Consent Statement

We verify that we have read the above letter and are giving permission for the ECE496 course coordinator to use our reports as outlined above.

Team #: 659 Project Title: Unsupervised Learning of EEG States

Supervisor(s): Genov Administrator: Burgwin

Name	Fatima Siddiqui	Signature	Fatima Siddiqui	Date:	April 6
Name	Khalil Scott	Signature	Khalil Scott	Date:	April 6
Name	Mehak Kalra	Signature	Mehak	Date:	April 6
Name	Mina Assaad	Signature	[illegible]	Date:	April 6

Executive Summary (M. Kalra)

Clinicians researching seizures and epilepsy often spend significant time in cleaning the data as they must annotate the Electroencephalography (EEG) data to differentiate epileptic and normal brain activity. To reduce the time spent labelling the data, the team was given the task of automating the process using machine learning (ML). The team solved this problem by taking three different approaches and choosing the one with the best result as the final design. The three different methods that were implemented were convolutional neural networks, stacked autoencoders, and clustering.

This document overviews the three prototypes with their system and module level descriptions. All these designs were then assessed based on the requirements the team had laid out. The two most important requirements taken into consideration were that the design should generate correct labels for classifying input data between seizure and non-seizure and should label data faster than signal sampling rate. Moreover, the team tested the prototypes based on the criteria above to select the final design that performed the best at labelling data. This report highlights the output received after the testing procedure and displays the signal features of the designs.

After conducting the tests the team found that the clustering algorithms show the most promising results, passing most of the required tests. Within clustering, the Fuzzy C-means (FCM) algorithm had the best overall performance. FCM achieved the highest average sensitivity at 82% over the entire dataset. FCM also had the second highest labelling rate at 11262 Hz compared to the other two algorithms, K-means and BIRCH (balanced iterative reducing and clustering using hierarchies).

In conclusion, this report has provided a detailed overview of the team's efforts to automate the process of annotating EEG data for seizure detection. The FCM clustering algorithm is the only ML approach that passed the requirement of having a sensitivity greater than 80%, while at the same time passing the requirement of labelling the EEG segments at a rate greater than 256 Hz. The team hopes that the insights and recommendations presented in this report will prove useful for researchers and clinicians working in the field of epilepsy and seizure detection.

Mehak Kalra's Individual Contribution

As a member of the research project team, Mehak has made valuable contributions in several areas. Mehak's work began with researching preprocessing techniques and conducting a literature review on EEG signals, preprocessing, NWB and EDF files.

She then worked on converting EEG signals into valuable inputs for machine learning models. She also collaborated with Fatima to plan the Convolutional Neural Network approach and coded the CNN model while training it with various hyperparameters. This aspect of the project was particularly challenging, but it was also incredibly rewarding to see the results that the team achieved.

Another area where she contributed was in developing the Implementation Plan Document, which involved dividing the work among the team members and ensuring that they stayed on track with the team's goals. Additionally, she assisted in developing the testing document, outlining the requirements for potential designs, and working on the visual interface, which required her to pay close attention to detail. She created and presented the introduction, motivation, problem and diagnosis slides in the oral presentation. Mehak was also responsible for the frontend of the project i.e. making a desktop application.

Throughout the project, Mehak learned a lot about teamwork, communication, and project management. She also gained valuable technical skills, particularly in data analysis and coding. Most importantly, she gained a deep appreciation for the research process, and is excited to continue working on research projects in the future.

Fatima Siddiqui's Individual Contribution

Fatima Siddiqui initiated the project by importing EDF signals onto Jupyter notebook for preprocessing, including transforming the signals into 1-D continuous wavelet transforms using the Morlet mother wavelength and splitting the data into train, test, and validation sets. She designed and trained a CNN model using a single EDF data input, obtaining training, validation accuracies/losses, and test accuracy for a single EDF file with hyperparameters such as batch size, epochs, and learning rates. Fatima then trained the CNN model for a batch size greater than one and with the entire dataset.

Moreover, Fatima contributed to the project by working on the test report, collaborating with Khalil on integrating the scalogram padding code into the under-sampling code, adding padding to the scalograms, and updating the CNN training function with the new data processing method. She also incorporated the under_sampling code into the CNN architecture and improved the architecture by adding more convolution and pooling layers.

Fatima also helped to prepare an oral presentation that efficiently communicated their project's results to the audience. This involved organizing the presentation's content in a clear and concise manner, creating visual aids such as slides and graphs to illustrate the results, and practicing the delivery to ensure a smooth and confident presentation.

Furthermore, Fatima was responsible for designing the desktop application's frontend used to display the analysis results. She conducted research on design concepts and user interface patterns appropriate for their project and collaborated with the team to finalize the design.

Khalil Scott's Individual Contribution

Khalil Scott was primarily responsible for researching and implementing the clustering approach. He researched Digital Signal Processing methods for transforming EEG signals into a format that could be handled by traditional clustering algorithms, such as K-means. Khalil wrote the code to train and test the K-means, Fuzzy C-means, and BIRCH algorithms. He also wrote the code to perform multilevel discrete wavelet transformation on EEG signal segments and calculate the segment's entropy, standard deviation, average power, and mean.

Khalil wrote the code to extract labels from the CHB-MIT dataset using regular expressions. He also wrote the code to split EEG signals into 5-second segments using a sliding window with 90% overlap. Khalil wrote the code to perform undersampling to balance the number of non-seizure and seizure examples in the dataset, as well as the code to evaluate the quality of the clustering algorithms.

For the clustering algorithm, Khalil implemented cross-validation testing, evaluating the performance of the algorithms on a single file. To evaluate the algorithms, Khalil wrote code to calculate their detection latency, sensitivity, and false positive rate. To save time on preprocessing, Khalil saved the preprocessed data and their labels in JSON files for faster reloading. This drastically reduced the training and testing of the clustering algorithms.

Khalil did minor work on the stacked autoencoder and CNN approaches as well. He helped write the testing code for the two approaches and helped with bug fixes. He also wrote a different version of the undersampling code to reduce the amount of RAM required to train the stacked autoencoder and CNN using scalograms.

Mina Assaad's Individual Contribution

Mina Assaad was primarily responsible for researching, developing, training, and testing the stacked autoencoder. He experimented with different techniques found in research such as dynamic creation of stacked autoencoder layers, freezing encoder weights, and transferring saved parameters to new neural network architectures. Mina also wrote, debugged, and tested the training code for the two training stages of the stacked autoencoder, which were the unsupervised pre-training and supervised fine-tuning stages.

Mina also implemented a technique for loading data from the European Data Format (EDF) files one batch at a time which solved a memory problem that the team was experiencing when attempting to load the entire dataset before training. This also allowed the team to use CUDA compatible GPUs without running out of VRAM which significantly reduced training times.

A large bulk of the work here was experimenting with different hyperparameters, which involved determining the optimal number of encoder layers, their output channels and stride sizes, batch sizes, number of epochs, learning rate, which activation functions to use, number of linear layers, and the number of neurons they included.

The task of continuing the training and testing of the convolutional neural network (CNN) was handed to Mina from Fatima toward the end of February. This primarily involved incorporating the per-patient training methodology that was adopted based on supervisor feedback in which multiple CNN's are trained on one patient each for all of the twenty-three patients in the CHB-MIT dataset. Per-patient training and testing was also done for the stacked autoencoder.

Mina also wrote the testing code for both the stacked autoencoder and CNN models which consisted of testing the sensitivity, latency, false positive rate, and labelling rate. A lot of this code was borrowed from Khalil's testing code for clustering but was modified to correctly load and input the data properly for the stacked autoencoder and CNN. He also wrote additional code to automate overnight testing for all twenty-three patients for both of these models. Additionally, Mina wrote the code for generating and outputting the scalogram and saliency heat map images for the CNN and stacked autoencoder, as well as saving data points to text files for clustering.

1.0 Introduction (author: M. Kalra)

As a part of the fourth year design course ECE496, the team completed a year-long project titled 'Unsupervised Learning of EEG States for the Application of Seizure Detection in Epilepsy', which is summarized in this document. This report provides an overview of the motivation, design, experimentation, testing, and final assessment of the three machine learning (ML) approaches researched in this project. The report ends with a final conclusion and reflection.

1.1 Background and Motivation (author: M. Kalra)

Researchers and clinicians studying seizures and epilepsy often spend a significant amount of time cleaning data, as they must annotate Electroencephalography (EEG) data to distinguish epileptic activity from normal brain activity. The process of annotating data is time-consuming and can take several hours per patient. To reduce the time spent by clinicians labelling the data, the team was given the task to automate the process. However, EEG data can be very complex because the signals are nonlinear, constantly changing, and noisy, which makes it difficult for ML algorithms to label the data [1]. The wave-like characteristics of EEG data mean there are multiple varieties of preprocessing methods that could either prioritize its frequency, periodicity, or both [2]. To address these challenges, the team has developed a software toolbox that employs machine learning (ML) techniques in conjunction with digital signal processing (DSP) to preprocess unlabelled EEG data and integrate it effectively.

1.2 Project Goals and Requirements (author: M. Kalra, M. Assaad)

The goal of this project was to experiment and arrive at the best ML algorithm based on their performance in labelling accuracy and labelling speed, which will be used as the final solution in the software toolbox. The team investigated three different machine learning (ML) approaches for labeling EEG data. The first method involved using a convolutional neural network (CNN) based on deep learning. The second approach utilized a convolutional stacked autoencoder. The third method involved using clustering algorithms, such as K-means and BIRCH (balanced iterative reducing and clustering using hierarchies).

Changes were made to the requirements presented in the Testing Document based on requests made by the supervisors. Requirement 4 was changed from a function to an objective. The labelling accuracy is now measured by two requirements: requirement 1 that measures sensitivity and requirement 3 that measures detection latency. Sensitivity measures how many seizure intervals, not segments, the ML approach was able to detect in a given EEG record. This provides a more meaningful measurement of generating correct labels as it shows how many seizure intervals the model was able to detect. The detection latency refers to the delay between when a seizure actually starts in a seizure record (an EEG record containing at least 1 seizure) and when the model predicts the seizure starts. The team decided to prioritize requirement 1 as a low sensitivity indicates that the model cannot accurately detect seizures. The team also prioritized requirement 2 as the model must annotate EEG signals faster than their sampling rate of 256 Hz. This is to prevent a bottleneck as EEG signals are constantly fed into the software toolbox. Table 1 outlines the latest revision of the project requirements that were tested.

Table 1. Requirements for Testing

ID	Requirement	Type	Pass/Fail Criteria
1	Generate correct labels for classifying input data between seizure and non-seizure	Function	Sensitivity must be greater than 80% [3]
2	Label data faster than signal sampling rate	Function	Must label more than 256 samples per second (each segment has 1280 samples) [3]
3	Labelled seizure start time should be close to the clinical onset of seizure	Objective	Detection latency between labelled seizure start time and clinical onset should be 20 seconds [4]
4	The model should have a high sensitivity	Objective	Sensitivity should be 100%
5	Provide clinical insights into how labels are determined	Objective	Display signal features for all segments

6	Must accept NWB files as input	Constraint	Signal extraction and labelling run with zero errors

2.0 Final Design (author: F. Siddiqui, K. Scott, M. Assaad)

This section provides an overview of the final design of each of the three ML approaches.

2.1 System-level Overview (author: K. Scott)

The software toolbox will require a number of steps to implement fully. First, a user will import a European Data Format (EDF) file using the toolbox's GUI. The EDF file contains EEG records that are extracted using Python. Feature extraction is performed on the input signals in order to reduce their complexity. Once a set of features has been extracted, an ML algorithm will be used to annotate the data as normal brain activity or epileptic seizure activity. The labels will then be displayed on the screen.

2.2 System Block Diagram (author: K. Scott)

A high-level block diagram detailing the workflow of the software toolbox is shown in Figure 1.

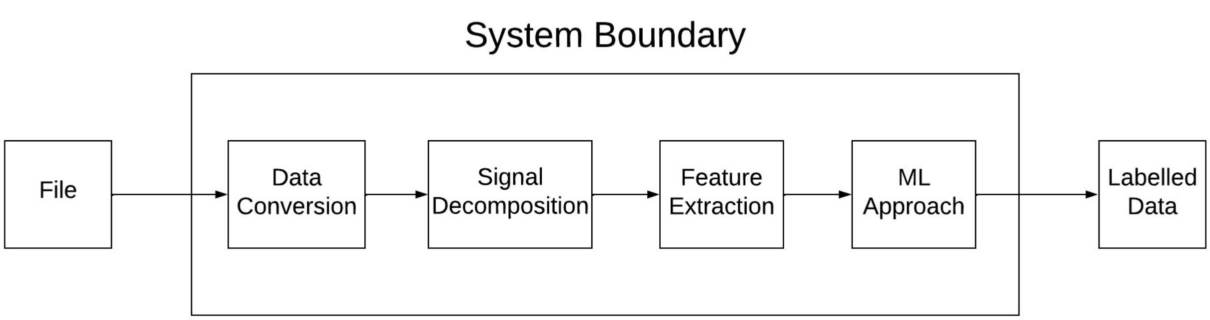


Figure 1. High-level block diagram of the software toolbox.

2.3 Module-Level Descriptions and Designs (author: F. Siddiqui, K. Scott, M. Assaad)

The team investigated three different ML methods for labelling EEG data. The first approach involved deep learning utilizing a CNN. The second approach involved a convolutional stacked autoencoder. Lastly, the team explored the use of clustering algorithms, such as K-means. In the following subsections, an overview of each approach will be presented. Tables 2 and 3 describe the modules of the software toolbox. Table 2 describes the first two modules, Data Conversion and Signal Decomposition, which are the same for all three models. The last two modules, Feature Extraction and ML Approach, differ for each model and are described in Table 3.

Table 2. First Two Modules of the Software Toolbox

Shared Modules				
Module	**Input**	**Output**	**Function**	**Evaluation**
Data Conversion	EDF/NWB files	Array of EEG signals	Read EEG signals from a file and format them for use in Python.	N/A
Signal Decomposition	Array of EEG signals	Array of EEG wavelet coefficients per signal	Perform digital signal processing functions on the EEG signals to obtain the wavelet coefficients.	N/A

Table 3. Last Two Modules of the Software Toolbox for each ML Approach

CNN				
Module	**Input**	**Output**	**Function**	**Evaluation**
Feature Extraction	Array of EEG wavelet coefficients	Array of scalograms	Array of wavelet coefficients is resized into an image, which is the scalogram.	N/A
ML Approach	Array of scalograms	Data labels	Use the CNN's convolution layers to extract the features from the scalogram images and input them into the linear layers to label the data.	Compute the detection latency, sensitivity, and false positive rate
Stacked Autoencoder				
Module	**Input**	**Output**	**Function**	**Evaluation**
Feature Extraction	Array of EEG wavelet coefficients	Array of scalograms	Array of wavelet coefficients is resized into an image, which is the scalogram.	N/A

ML Approach	Array of scalograms	Data labels	Use the stacked autoencoder's encoder layers to extract the features from the scalogram images and input them into an attached classifier to label the data.	Compute the detection latency, sensitivity, and false positive rate
Clustering				
Module	**Input**	**Output**	**Function**	**Evaluation**
Feature Extraction (Clustering)	Array of EEG wavelet coefficients	Array of signal features	Compute the average power, mean, standard deviation, and entropy of each signal to use as the input features for the cluster algorithm.	N/A
ML Approach (Clustering)	Array of signal features	Data labels	Run the clustering algorithm on the extracted features and cluster them. The cluster assignments will be used to determine the label of the signal.	Compute the detection latency, sensitivity, and false positive rate

2.3.1 CNN (author: F. Siddiqui)

The CNN model was selected because it is the most popular deep learning method pursued to automate seizure detection [5]. A 2-Dimensional CNN architecture was used to preprocess 2-Dimensional Time-Frequency (2D-TF) representations, also known as scalograms, of the EEG data [5]. The scalograms were generated using the CWT [5], which were then inputted into the model. The CNN architecture consists of three convolutional layers using 3x3 kernels. Two pooling layers with 3x3 kernels were also used. A Rectified Linear Unit (ReLU) activation function was used after each convolution and pooling layer to introduce non-linearity to the model [5]. Following the preprocessing step using the CNN, three fully connected layers were used in the classification layer to label the signals as either seizure or non-seizure. ReLU was used as the activation function for each of these layers [5].

2.3.2 Autoencoder (author: M. Assaad)

The stacked autoencoder was an ML approach used in similar applications to this project's, making it a good candidate for experimentation. Similar to the CNN, the CWT was used to transform the incoming signals into segmented scalogram images which were passed as input into the stacked autoencoder model. Unsupervised training was performed to improve the encoder layers' performance on extracting features. This was done by minimizing the loss between the encoded input signal segments and decoded output signal segments. Supervised training was then performed by replacing the decoder layers with one linear layer for performing the classification, and fine-tuning the weights of that layer.

2.3.3 Clustering (author: K. Scott)

The team decided to test implementations of the K-means, Fuzzy C-means (FCM), and BIRCH clustering algorithms for labelling. Since clustering algorithms cannot process raw signals due to their high complexity, the discrete wavelet transform (DWT) function was used to decompose the signal and obtain its wavelet coefficients. Using the wavelet coefficients, the average power, mean, standard deviation, and entropy of the signal were calculated [6]. These values were the input features for the clustering algorithms. The clustering algorithms were trained and evaluated using the CHB-MIT dataset. Cross-validation testing was used, designating a single EEG record for testing and using the remaining EEG records for training. The clustering algorithms would make predictions on the testing record, annotating the EEG segments as seizure or non-seizure. Afterwards, the detection latency and sensitivity were calculated to evaluate labelling accuracy.

2.4 Assessment of Final Design (author: F. Siddiqui, K. Scott, M. Assaad)

The three models were assessed using requirements outlined in Table 1. The two important functional requirements considered were 1 and 3; they are defined in the second paragraph of Section 1.2, with objective requirements considered as desirable end goals. The following subsections discuss results acquired for each of the three models.

2.4.1 CNN (author: M. Assaad)

The CNN model failed in several of the requirements. Its average sensitivity of 55.5% was below the required minimum measurement. Its latency was high at 21.4 seconds from the seizure's actual start time. The model did manage to label at a rate higher than the minimum set out, labelling at an average of 150300 Hz. It displayed features for every signal segment, fulfilling requirement 4. The images in Figure 5 show saliency maps of different signal segments labelled by CNN. Specifically, Figure 5 (a) shows a segment correctly labelled as non-seizure, (b) shows incorrectly labelled as non-seizure, (c) correctly labelled as seizure, and (d) incorrectly labelled as seizure. The two saliency maps in Figure 5 (a) and (b) look similar to each other but different to those displayed in (c) and (d), the same also holds for (c) and (d). This indicates the model is doing well in classifying, given the features it sees, but features being extracted are not poorly representing the signal. Clearly, this model cannot handle data as complex as EEG signals. In future, more complex deep learning models like multi-layer perceptrons can be tested with to asses their capabilities in feature extraction of EEG signals and predictions.

2.4.2 Autoencoder (author: M. Assaad)

The stacked autoencoder was a slight improvement from CNN, achieving an average sensitivity of 60.5%. It also managed to acquire a very low average latency of 2.11 seconds, passing that objective requirement. It also labelled at a rate much higher than the minimum 256 Hz requirement, with an average of 156700 Hz. The stacked autoencoder is able to display features for every signal segment but suffers the same issue as before with the CNN, described above. This is illustrated by the saliency heat maps in Figure 6. Like CNN, experimenting with different transforms other than CWT in the future could benefit this research. Experimenting with other types of stacked autoencoders like dense autoencoders could potentially yield better results.

2.4.3 Clustering (author: K. Scott)

The Clustering algorithms shows most promising results, passing most tests, seen in section 3.1. Of the three clustering algorithms, FCM performs best in terms of accuracy, achieving an average sensitivity of 82% over the entire dataset. Compared to the other two clustering algorithms, FCM had the second lowest average detection latency at 5.67 seconds, and second highest labelling rate at 11262 Hz.

Figure 7 shows data points for four different signal segments classified by the BIRCH clustering algorithm. The data point values of the top two segments (true negative and false negative) have close values for average power and standard deviation, which are different from those same values for the bottom two segments (true positive and false positive). Those two segments in turn have similar values. As with the previous two ML approaches, this indicates the BIRCH clustering algorithm is struggling to extract features that properly characterize the EEG signals.

From testing results, partitioning algorithms such as K-means and FCM perform better overall when compared to BIRCH. K-means and FCM, produce very little false negative and false positive data points, indicating they are performing better in terms of labelling. With more time, the team would pursue further exploration of other partitioning algorithms. K-medoids is one algorithm that is an improvement to K-means algorithm and could be researched in the future.

3.0 Testing and Verification (author: K. Scott, M. Assaad)

Since the creation of the Testing Document, there have been changes to the structure of training and testing each ML algorithm. Now, each model is trained on a per-patient basis, meaning only one patient from the CHB-MIT dataset is used to train and test each model one at a time. This is because each patient has different onsets of seizures, making the general patient training approach used previously no longer practical. While testing the stacked autoencoder, it was found that it takes approximately 2.5 - 3 hours to run testing on a single EEG record. As a result, each EEG record in the test set was shrunken to make testing significantly faster for all three models. A mean average was applied to the measurements obtained for each of the requirements. These averages are included in Table 4. It was also important to make sure the model functioned properly on a whole EDF file, so one additional test was performed on an entire EDF file for all three models. The results for these tests are included at the bottom of this chapter in Section 3.4.

3.1 Verification Table (authors: K. Scott, M. Assaad)

Table 4 outlines the test results for each requirement listed in Table 1. The testing procedures used to acquire these results are identical to the tests described in the Testing Document, except now the test set is split by patient into multiple test sets, with each patient test set of EEG records used to test the model trained for that patient. As stated previously, an average is taken on all of the results for the sensitivity, latency, and labelling speed measurements. Also, one additional test has been added to the test suite to measure latency in a similar fashion to testing sensitivity.

Table 4. Testing and Verification of Each Algorithm

ID	Requirement	Verification	Verification Result and Proof

		Method	CNN	Autoencoder	Clustering
1	Generate correct labels for classifying input data between seizure and non-seizure	TEST: Compute sensitivity	FAIL. 55.5%.	FAIL. 60.5%.	BIRCH: FAIL. 54% K-means: FAIL. 79% FCM: PASS. 82%
2	Label data faster than signal sampling rate	TEST: Measure labelling rate	PASS. 150300 Hz.	PASS. 156700 Hz.	K-means: PASS. 11098 Hz BIRCH: PASS. 11336 Hz FCM: PASS 11262 Hz
3	Labelled seizure start time should be close to the clinical onset of seizure	TEST: Compute detection latency	FAIL. 21.4 s.	PASS. 2.11 s.	BIRCH: PASS. 8.85s K-means: PASS. 4.98s FCM: PASS. 5.67s
4	The model should be accurate	TEST: Compute sensitivity, and false positive rate	FAIL. 55.5%.	FAIL. 60.5%.	BIRCH: FAIL. 54% K-means: FAIL. 79% FCM: FAIL. 82%
5	Provide clinical insights into how labels are determined	TEST: View feature output	PASS.	PASS.	PASS.

6	Must accept NWB files as input	TEST: Import and process NWB file	UNTESTED. Could not convert NWB to EDF files, but toolbox successfully accepts and processes EDF files

3.2 Raw Output from Testing (author: M. Assaad)

This section presents some of the raw output produced during the automated testing for each ML algorithm, and also goes over some of their key highlights.

3.2.1 CNN (author: M. Assaad)

Figure 2 shows the output produced by the automated CNN testing code for four patients. The first highlight to mention is that Figure 2 (a) shows the model was unsuccessful in detecting any correct seizure intervals, as the sensitivity shown is 0.0. This also means a value of -1 was returned for the latency because there were no detected seizure intervals to compute the latency on. Figure 2 (b) shows that all seizure intervals were correctly detected in this instance (a value of 1.0 indicates 100% sensitivity). All outputs indicate labelling speeds over 500 times higher than the 256 Hz requirement. The false positive rates (FPR) were additional scores that were kept track of during testing. An FPR of 921,600 (the number of samples in a given hour) indicates 100% of non-seizure segments were labelled incorrectly. The final averages of these results are highlighted in Table 4.

```
Test results for CHB04:

Average latency: -1
Average sensitivity: 0.0
Average FPR: 327965.6652360515
Average speed: 136075.78187406104
```

(a)

```
Test results for CHB05:

Average latency: 61.5
Average sensitivity: 1.0
Average FPR: 57949.21184825913
Average speed: 155358.72559759955
```

(b)

```
Test results for CHB07:

Average latency: -1
Average sensitivity: 0.0
Average FPR: 159183.46400570206
Average speed: 151068.45706305013
```

(c)

```
Test results for CHB10:

Average latency: 45.5
Average sensitivity: 1.0
Average FPR: 72057.87358421629
Average speed: 145657.49853895922
```

(d)

Figure 2. The test output produced for four patients by the automated CNN testing code.

3.2.2 Autoencoder (author: M. Assaad)

Figure 3 shows the output produced by the automated stacked autoencoder testing code for four patients. Figure 3 (a), (b), and (d) show the model is detecting 100% of the seizure intervals, with latencies much lower than the 20 second objective goal. In Figure 3 (c), half of the seizure intervals were detected. The latency in this case is only computed as an average of the latencies of seizure intervals that were successfully detected by the model, which is 0 seconds.

```
Test results for CHB01:

Average latency: 2.0
Average sensitivity: 1.0
Average FPR: 98932.82031550597
Average speed: 161819.53853938807
```

(a)

```
Test results for CHB03:

Average latency: 3.5
Average sensitivity: 1.0
Average FPR: 151219.05482041588
Average speed: 160788.209931364
```

(b)

```
Test results for CHB06:

Average latency: 0.0
Average sensitivity: 0.5
Average FPR: 347269.5652173913
Average speed: 150576.78443510592
```

(c)

```
Test results for CHB11:

Average latency: 0.0
Average sensitivity: 1.0
Average FPR: 244164.15584415582
Average speed: 154604.34799458613
```

(d)

Figure 3. The test output produced for four patients by the automated testing code for the stacked autoencoder.

3.2.3 Clustering (author: K. Scott)

Figure 4 shows output produced by the automated clustering testing code for a single patient. Both K-means and FCM detected 100% of the seizure intervals while BIRCH detected 86%. FCM had a detection latency more than 10 times smaller than BIRCH, and more than 2 times smaller than K-means.

```
CHB01
+---------+-------------+----------------------------+---------------------------------+
| Latency | Sensitivity | False Positive Rate (Ictal) | False Positive Rate (Interictal) |
+---------+-------------+----------------------------+---------------------------------+
|   1.5   |     1.0     |      58780.2746064522      |         84648.9529931231         |
+---------+-------------+----------------------------+---------------------------------+
```

FCM

CHB01

Latency	Sensitivity	False Positive Rate (Ictal)	False Positive Rate (Interictal)
3.4285714285714284	1.0	28625.113783886747	56833.28717329532

K-means

CHB01

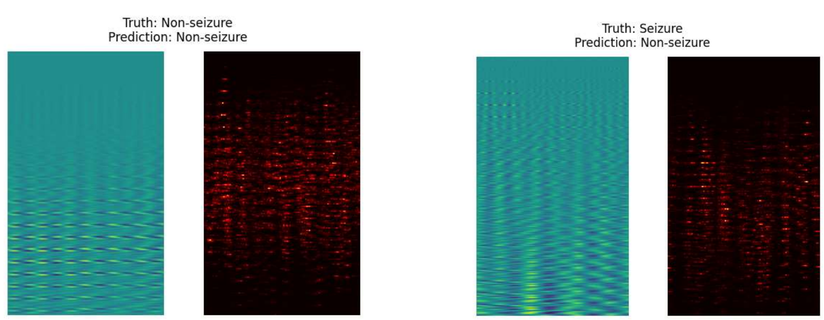
Latency	Sensitivity	False Positive Rate (Ictal)	False Positive Rate (Interictal)
19.0	0.8571428571428571	1835.4883567600334	12142.124302870046

BIRCH

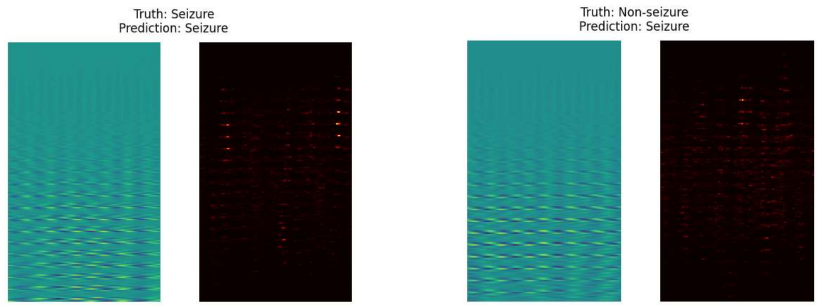
Figure 4. The test output produced for patient 1 by the automated testing code for all three clustering algorithms.

3.3 Displaying Signal Features (author: M. Assaad)

All three ML algorithms were successful in fulfilling Requirement 4 which required them to display features for all signal segments. The CNN and stacked autoencoder were both able to display these features in the form of scalogram images. Saliency heat maps were also produced to illustrate how the model used those features when making a decision. The clustering algorithms were also able to display features in the form of data points. Figure 5, 6, and 7 show examples of displayed features for the CNN, stacked autoencoder, and clustering, respectively.

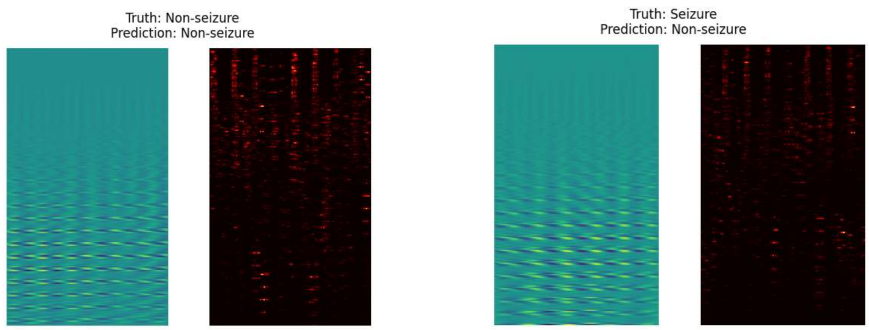


(a) (b)

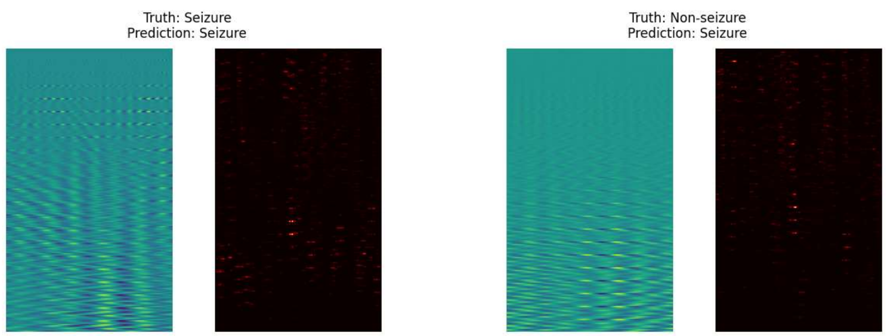


(c) (d)

Figure 5. Scalograms and saliency heat maps produced for each labelling case for the CNN.



(a) (b)



(c) (d)

Figure 6. Scalograms and saliency heat maps produced for each labelling case for autoencoder.

```
Truth: Non-seizure, Prediction: Non-seizure
[4848.1308945688925, -0.09500807319154572, 69.65556448613636, 7.150962199253181]
```

(a)

```
Truth: Seizure, Prediction: Non-seizure
[9249.082385185662, -0.710765902400358, 96.20696512602335, 7.140173916287267]
```

(b)

```
Truth: Seizure, Prediction: Seizure
[87937.1279538059, 1.22941454095522, 296.6548536200274, 7.15851399732932]
```

(c)

```
Truth: Non-seizure, Prediction: Seizure
[83514.82719644303, -1.9081783462288495, 289.0955307789596, 7.150962199253181]
```

(d)

Figure 7. Data points produced for each of the four labelling cases for the BIRCH clustering algorithm. Each of the four numbers from left to right represent average power, mean, standard deviation, and entropy.

3.4 Testing on a Full EDF File (author: M. Assaad)

As explained previously, all of the above sections were tested on shrunken versions of the EEG records to reduce testing time. However, to verify success on a full EDF file, since the selected algorithm would process an entire file in practice, each algorithm was run once on the chb01_03.edf file, using the version trained for patient 1 (CHB01). Figures 8, 9, and 10 show the output produced from running the tests on this file. In all cases every model managed to detect all seizure intervals. Latencies and labelling speeds also met the requirements.

```
Test results for CHB01_03:

Average latency: 0.0
Average sensitivity: 1.0
Average FPR: 245541.05912823055
Average speed: 157900.133217468
```

```
Test results for CHB01_03:

Average latency: 0.0
Average sensitivity: 1.0
Average FPR: 318479.96914090484
Average speed: 164948.12284969157
```

Figure 8. Test results for the CNN (left) and stacked autoencoder (right) on the full chb01_03.edf file.

BIRCH

Latency	Sensitivity	False Positive Rate (Ictal)	False Positive Rate (Interictal)
17.0	1.0	756.1360004408442	0

K-means

Latency	Sensitivity	False Positive Rate (Ictal)	False Positive Rate (Interictal)
7.0	1.0	16685.777263459524	0

FCM

Latency	Sensitivity	False Positive Rate (Ictal)	False Positive Rate (Interictal)
7.0	1.0	16533.42150217667	0

Figure 10. Test results for the three clustering algorithms on the full chb01_03.edf file.

4.0 Summary and Conclusion (author: M. Kalra, F. Siddiqui)

This project aimed to address the challenge of clinicians and researchers spending significant amounts of time annotating EEG data to differentiate epileptic and normal brain activity. The team was tasked with developing a solution that would significantly reduce the time and effort required for this process by using ML.

To achieve this goal, the team implemented three different approaches, including Deep Learning, Clustering, and Stacked Autoencoders. The team evaluated each prototype based on its ability to meet the project requirements, including generating correct labels for classifying input data between seizure and non-seizure and labeling data faster than the signal sampling rate.

The team's thorough testing and validation process provided substantial evidence for the effectiveness of the final design in addressing the original challenge of automating the process of annotating EEG data for seizure detection. Through their validation and acceptance tests, the team successfully demonstrated that the final design met most of the project goals and requirements, including generating accurate labels for classifying input data between seizure and non-seizure and labeling data faster than the signal sampling rate.

The key conclusions drawn from this project are that the Clustering algorithms, and in particular the FCM algorithm, are a promising approach for automating the process of annotating EEG data for seizure detection. This project's work has significant potential in the fields of epilepsy and seizure detection, as it can significantly reduce the time and effort required for clinicians to label EEG data, improving the efficiency and accuracy of diagnosis and treatment in these fields.

Future work on this project could explore the potential of combining the FCM algorithm with other machine learning techniques to further improve the accuracy and speed of the labeling process. Additionally, the team suggests that further work could be conducted to explore the potential applications of their work in other areas of research and industry where EEG data is used.

5.0 References

[1] Dai C;Wu J;Pi D;Becker SI;Cui L;Zhang Q;Johnson B; “Brain Eeg Time-series clustering using maximum-weight clique,” IEEE transactions on cybernetics. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32149677/. [Accessed: 24-Oct-2022].

[2] S. Scholl, “Fourier, gabor, Morlet or wigner: Comparison of time-frequency transforms,” arXiv.org, 17-Jan-2021. [Online]. Available: https://arxiv.org/abs/2101.06707. [Accessed: 24-Oct-2022].

[3] (2022, July 15). Re: Interest in Project for ECE496.

[4] J. Koerner, “Machine Learning-Driven Patient-Specific Early Seizure Detection for Neuromodulation Devices” M.S. thesis, Dept of ECE, UofT, 2020. [Online]. Available: https://tspace.library.utoronto.ca/bitstream/1807/103390/1/Koerner_Jamie_Michael_202011_MAS_thesis.pdf

[5] X. Wang, T. Ristaniemi, and F. Cong, “One and two dimensional convolutional neural networks for seizure detection using EEG signals,” *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021.

[6] C. K. Pradhan, S. Rahaman, M. Abdul Alim Sheikh, A. Kole, and T. Maity, “EEG signal analysis using different clustering techniques,” *SpringerLink*, 01-Jan-1970. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-13-1498-8_9. [Accessed: 05-Jan-2023].