Synopsys 2024 Project Notebook

Applying Unsupervised Learning Techniques to Automated EEG Artifact Removal and Seizure Detection

Dylan Dale

Journals/Planning:

Background research notes:

Reducing Noise in EEG Data

Look into

- Pre-existing artifact removal methods
    - ICA
- PCA

Research Purpose

- EEGs are applied widely but are mainly limited in their susceptibility to noise
- Effective removal of artifacts and noise can be beneficial in diagnoses
- Epilepsy affects 50m+ people worldwide, 3.4m in the US, one of most common neurological conditions
- Diagnosis is potentially life-altering, misdiagnosis can be detrimental to health
- High misdiagnosis rate of ~20%
- Applications to other conditions, BCIs

Proposed plan

- Standard de-noising/bandpass filter
- Apply matched filter theorem or some method of optimizing signal-to-noise ratio? (if feasible)
- Write ML model to identify/classify components of signal as artifacts
    - Supervised: SVM, k-NN, RNN, LSTM, CNN (first 2 likely to produce better results?)
    - Unsupervised: Not as widely explored
        - Better for clustering, increases flexibility because labeled training data is unnecessary

- Use CHB-MIT database for data

Matched filter theorem:

- To design a filter that has the strongest response to a particular signal v(s), in the sense of having the largest signal-to-noise ratio, design it so the transfer function H(s) “has the same shape” as V(s).

Filters:

- Smoothing filter
- Lowpass, highpass
- bandpass, notch
- Read: Digital Filters by Hamming

Applications of FFT:

- Convolutions, filtering kernels
- Interpolation of signals?

Consider use of:

- PCA, CNNs
- linear time invariant systems
    - “LTI systems are used extensively in the study of communications systems, where a fundamental concern is to distinguish the signal from noise and to design a filter that will do the job.”
- Elliptic Fourier analysis

Topic: **ML model to classify/remove EEG artifacts and perform seizure detection**

11/24/23:

After some background research on FFT and signal processing, decided upon a project involving EEG analysis.

11/26/23:

Looked into methods of artifact classification.
http://pe.org.pl/articles/2012/11a/50.pdf

Found article below which contains examples of possible datasets for validating artifact reduction. At least two separate datasets would be helpful, one for artifact removal and one for seizure detection. The CHB-MIT database will likely be used for the latter.
https://www.nature.com/articles/s41597-022-01524-x
12/4/23:

Researched potential models to be used. Autoencoders seem promising, as they can reduce the dimensionality of data to encapsulate only the most important parts of the data, which can then be used for clustering.

12/15/23:

Started work on registration forms. Plan to submit for signature within the next week so experimentation can be started.

12/18/23:

Learned about variational autoencoders, a modification of the autoencoder. Both could be tried to compare results between the two; the more accurate one would be used. Prior research has seen success with the VAE (https://openreview.net/forum?id=TVjLza1t4hI). Also considering using liquid-time constant networks for the seizure detection portion. They are a relatively new type of model, but have shown promise with time series modeling, which fits the goal of the project.

1/15/24:

Official start of experimentation. Data was obtained from CHB-MIT database for basic analysis, preprocessing and plotting. The data is bandpass filtered within the range 0.5 to 40 Hz; frequencies out of this range are unlikely to be related to brain activity. Could potentially try ICA to separate data into independent components for further analysis.

1/19/24:

Implemented autoencoder model for baseline testing. Applied model to the MNIST database to to validate performance on a basic example dataset. The model performed well on the handwritten number reconstruction; next steps will translate this to the EEG data. Considering using a variational autoencoder instead,

1/23/24:

Implemented variational autoencoder. Tested on the MNIST database, saw good performance.

Note listed on published paper: “The correct definition of "generative temperature" is not published yet, but used extensively in experiments… When worked out correctly, the beta does not multiply the generative error, as in this paper, so it would be very interesting to repeat the toy experiments here with the "correct" physical model instead.” This may be something to look into in the future.

1/27/24:

Updated autoencoder and variational autoencoder models to use convolutional encoder layers instead of linear layers. Retrained the models on the MNIST database, and saw slightly better results than before the update.

1/28/24:

Wrote code to segment EEG data into pieces that can be passed into models. Tested autoencoder and variational autoencoder on EEG data. The models performed poorly, likely an error in preprocessing. Plan to fix this soon.

1/30/24:

Data was being normalized per segment, causing disparities with normalization within a single EEG file. Data loading algorithm was rewritten to normalize the data for an entire file simultaneously, which caused the algorithms to work much better.

2/1/24:

Implemented the LTC model, which originally attempted to fit the curve with a straight line. More neurons were added and the model performed significantly better.

2/5/24:

The VAE was being trained with segments of length 0.25 s for testing purposes, and the LTC on segments of length 1 s. Updated these to 1 s and 5 s respectively, as these values make much more sense in context. After the update, training took about 50 minutes on CPU for the VAE (10 minutes on GPU) and about 30 minutes on CPU for the LTC (7 minutes on GPU). Combined, training takes about 75 minutes, which is well within the limit set by engineering goals.

2/10/24:

Implemented some methods of visualization such as scatter plots, and generated visualizations to use for the project board.

2/11/24 onward:

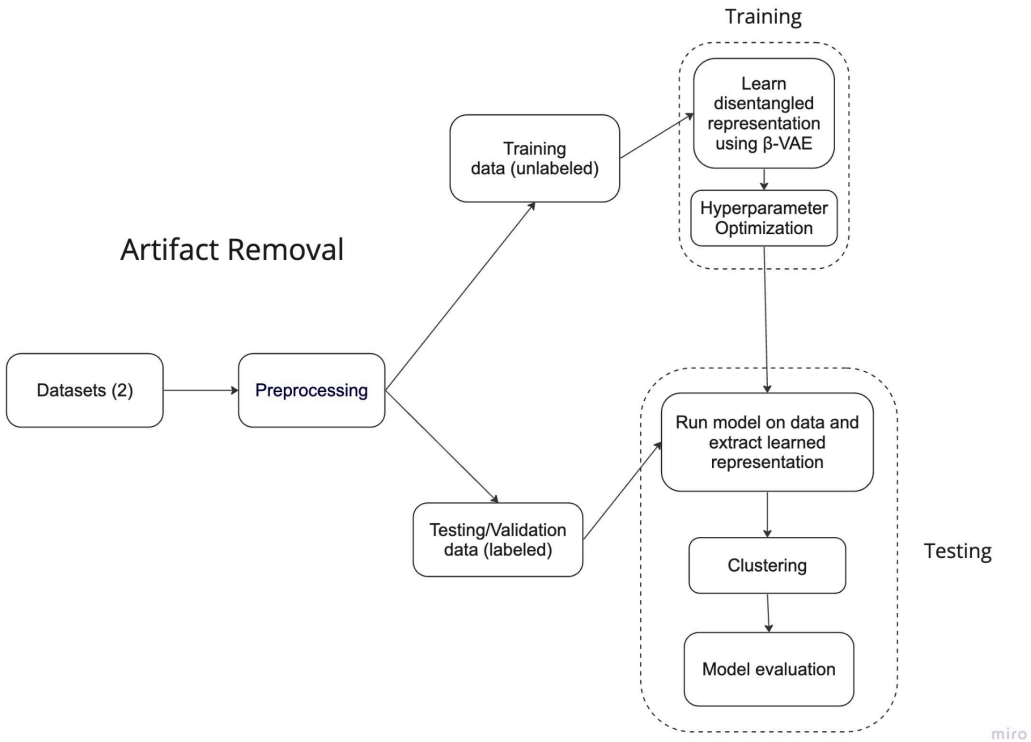
Worked on finalizing results and visualizations and preparing the project board and speech.

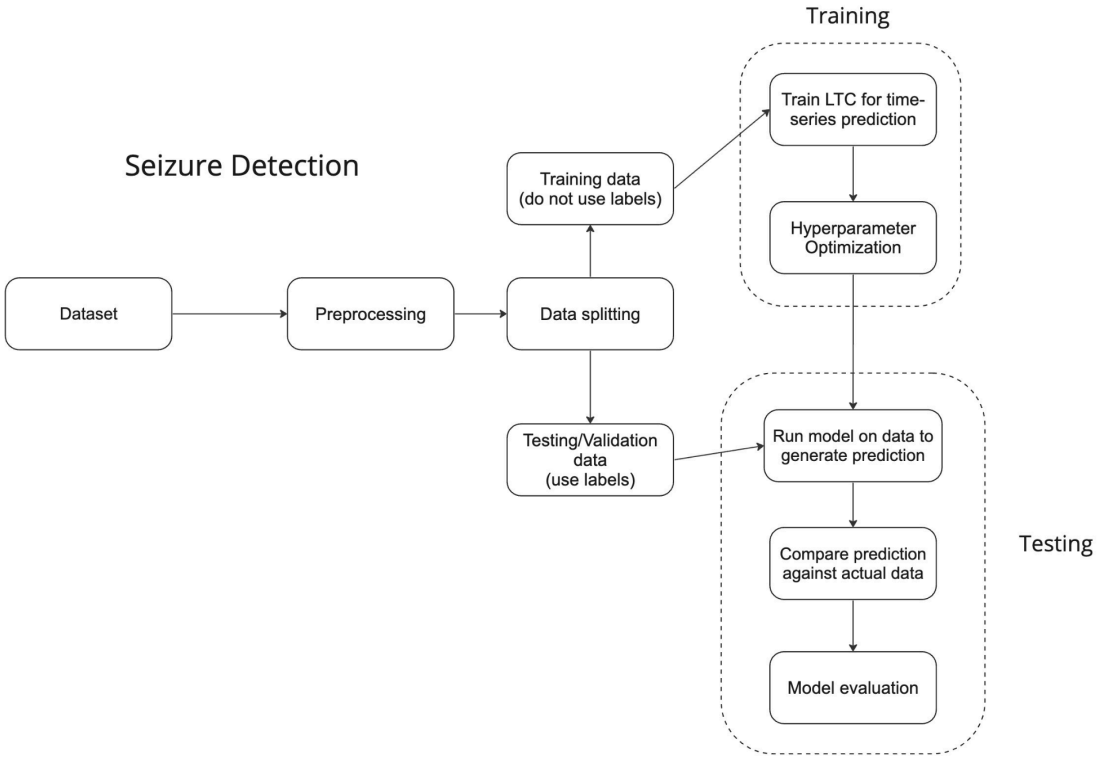
Project Description

Epilepsy is a very common neurological disorder, affecting around 50 million people worldwide. Epilepsy is usually diagnosed with EEG, but this requires extensive analysis by trained professionals. EEG data is highly susceptible to noise because of how weak brain activity is when picked up by electrodes on the scalp. In addition, brain activity can look very different across patients; one person's seizure activity might look like another person's normal activity. Due to these limitations, while significant progress has been made in the field of automated EEG analysis, the proposed methods have yet to be widely used in actual medical settings. In addition, many machine learning-based methods require labeled data, which can be hard to obtain and may not be representative of all patients. As such, accurate unsupervised methods for artifact removal and seizure detection, two major aspects of epileptic EEG analysis are very important.

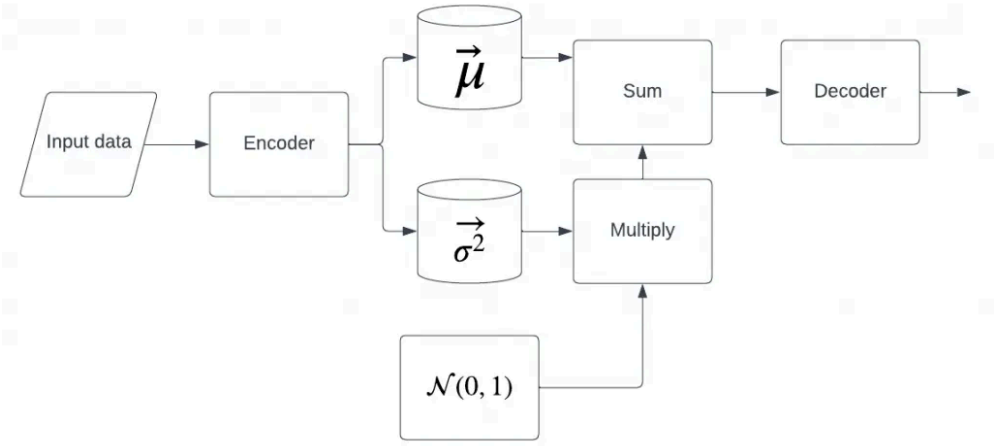
Brain activity tends to be much smaller than regions with artifacts, as artifacts may take place with the machinery or closer to the electrodes (as is the case with blinks). As a result, artifact removal algorithms typically rely on identifying outliers within the data. On the other hand, there are two main methods of seizure detection: a machine learning-based approach and a "biology"-based approach. The first kind attempts to apply ML and deep learning models to discover abnormalities within the data, while the second kind relies more on knowledge of signs that appear in the data prior to and during seizure onset. One example of the second kind is counting the number of major spikes occurring in the data, and triggering when the number crosses a certain threshold. This project chooses to employ a machine learning-based approach with minimal reliance on biological factors. This also makes it a good candidate for application to conditions other than epilepsy. While other groups have taken similar approaches in the past many of these attempts use supervised learning models, requiring data labeled with seizures. This poses a severe limitation that this project attempts to remedy, because labeled data requires careful analysis to generate and is therefore hard to come across.

For this project, I present a novel unsupervised learning pipeline applied to the CHB-MIT Scalp EEG database to achieve the goals described above. While the dataset has labels, these labels are not used for training. A variational autoencoder (VAE) is used to generate low-dimensional embeddings of EEG data that can be clustered to detect and subsequently remove abnormal activity. The newly proposed liquid time-constant network (LTC) is trained to model the artifact-free data and identify seizures. The project uses pytorch for machine learning, MNE for EEG data manipulation, and the ncps and pytorch-lightning libraries for formulation and training of the LTC model. Flowcharts are listed below:


Training
Learn disentangled representation using β-VAE
Hyperparameter Optimization
Training data (unlabeled)
Artifact Removal
Datasets (2)
Preprocessing
Testing/Validation data (labeled)
Run model on data and extract learned representation
Clustering
Model evaluation
Testing
miro


Training
Train LTC for time-series prediction
Hyperparameter Optimization
Seizure Detection
Training data (do not use labels)
Dataset
Preprocessing
Data splitting
Testing/Validation data (use labels)
Run model on data to generate prediction
Compare prediction against actual data
Model evaluation
Testing

The variational autoencoder (VAE) model is a neural network-based probabilistic method of non-linear dimensionality reduction. The model passes input data through a encoder network to produce a low-dimensional representation of the data, and then passes this representation back through a decoder network to recreate the original input data. For this project, convolutional encoding layers were used. In a VAE, the latent dimension is expressed as a probability distribution which is accomplished using two additional layers representing the mean and variance of the distribution.



The loss is calculated with a combination of mean squared error and KL Divergence (KLD), which measures the difference between a sampling from the latent distribution and the standard normal distribution to ensure the encodings remain close together. However, unlike other groups who have taken a similar approach, I decided to weight the mean squared error more than the KLD, as the goal of having a continuous latent space does not fit the objective. Having a continuous latent space is important when using VAE as a generative model, which is not the case here. The main reason for using a VAE over the AE is the superior expressivity it has. The loss is formulated like so, where beta is a hyperparameter greater than 1 that adjusts the weight of the different loss components. Making beta larger makes the model prioritize accurate reconstructions more during training.

$$Loss = \frac{1}{2}\sum_{j=1}^{J} 1 + \sigma_j^2 - \mu_j^2 - \ln \sigma_j^2 + \beta * \frac{1}{L}\sum_{l=1}^{L}(Y_l - \hat{Y}_l)^2$$

The liquid time-constant (LTC) network is a type of neural ordinary differential equation (neural ODE) model trained as a recurrent neural network (RNN) with the formulation shown below. The neural ODE is not a neural network in itself, it is simply a type of model where the hidden states are described as ordinary differential equations. It can be trained as various different types of models, like RNN here or VAE as the original authors propose. The LTC is modeled after the nematode C. elegans, which demonstrates complex behavior despite having only 302 neurons compared to the billions that humans have. Similar to the organism, the LTC model serves a simple yet performant alternative to more complex RNNs such as the LSTM, which require

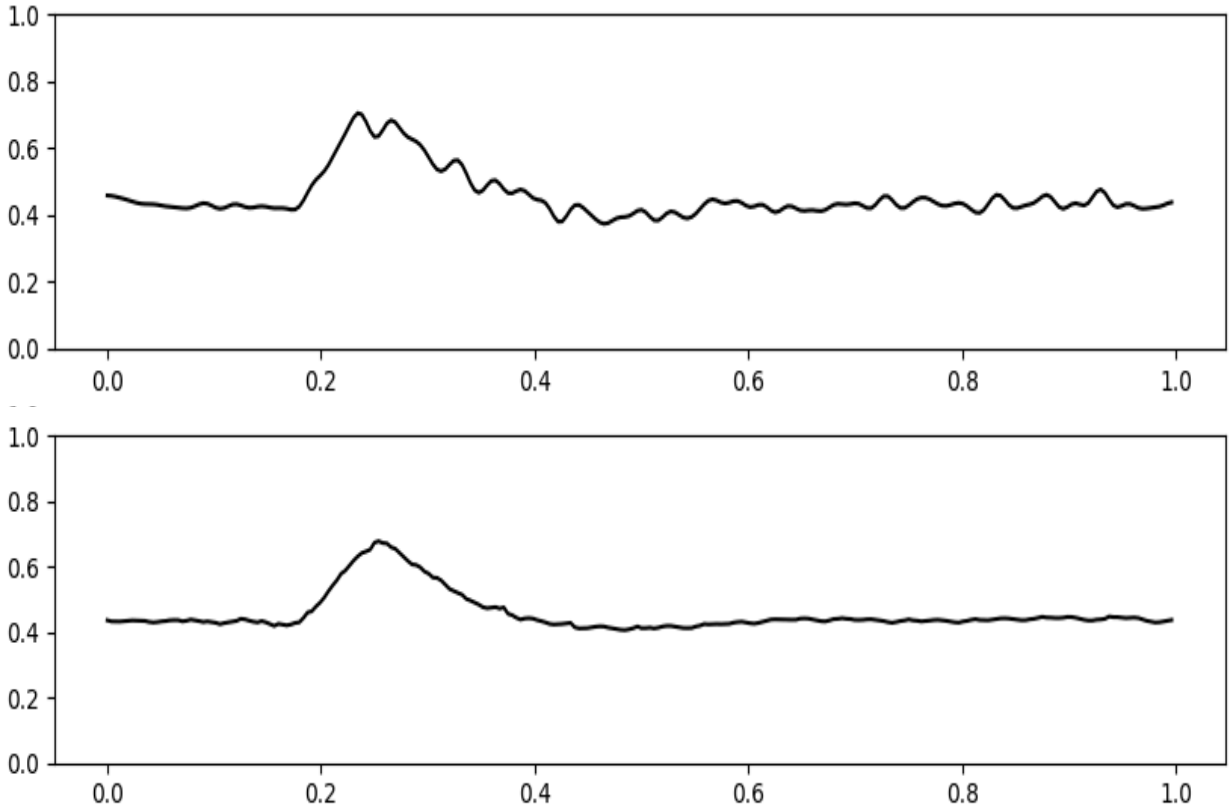
multiple orders of magnitude more neurons to function. My model only uses 100 neurons to fit the EEG data. The LTC describes the hidden states as ODEs like other neural ODE based models, but the formulation is quite different, as shown below.

$$\frac{d\mathbf{x}(t)}{dt} = -\left[\frac{1}{\tau} + f(\mathbf{x}(t), \mathbf{I}(t), t, \theta)\right]\mathbf{x}(t) + f(\mathbf{x}(t), \mathbf{I}(t), t, \theta)A.$$

Like other neural ODE models, the model is trained by solving the ODEs with some form of numerical ODE solver. The original authors use Euler's method for this purpose, but other methods are possible as well.

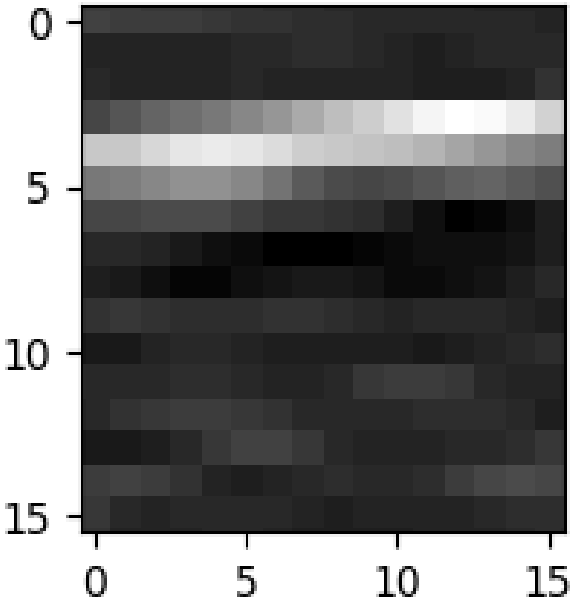
After training the LTC, the data is passed through the model for fitting. Due to the high ratio between normal activity and seizure activity, the model will learn the former much better than the latter. Thus, comparing the LTC output to the original data and observing the variance between the two is a feasible method of detecting seizures. For the purposes of this project, a detection threshold of 50% element-wise mean percent error was used, but this value can be optimized further for better performance.

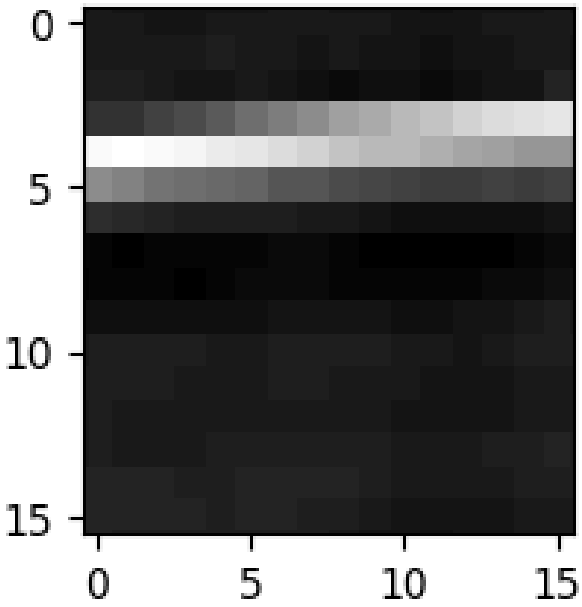
Visualizations and results are below:

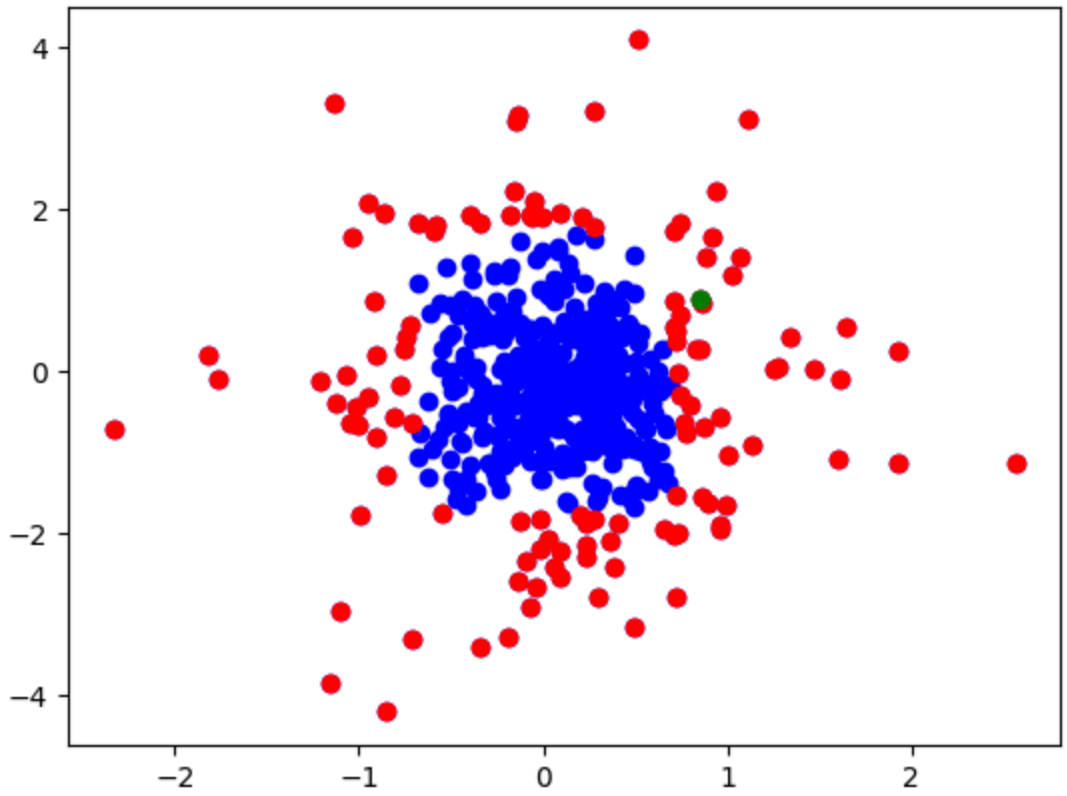


The plots above are 1 s long (256 length) and show an EEG segment containing an artifact. The large spike is not a normal occurrence in brain activity, and may be something like a blink.

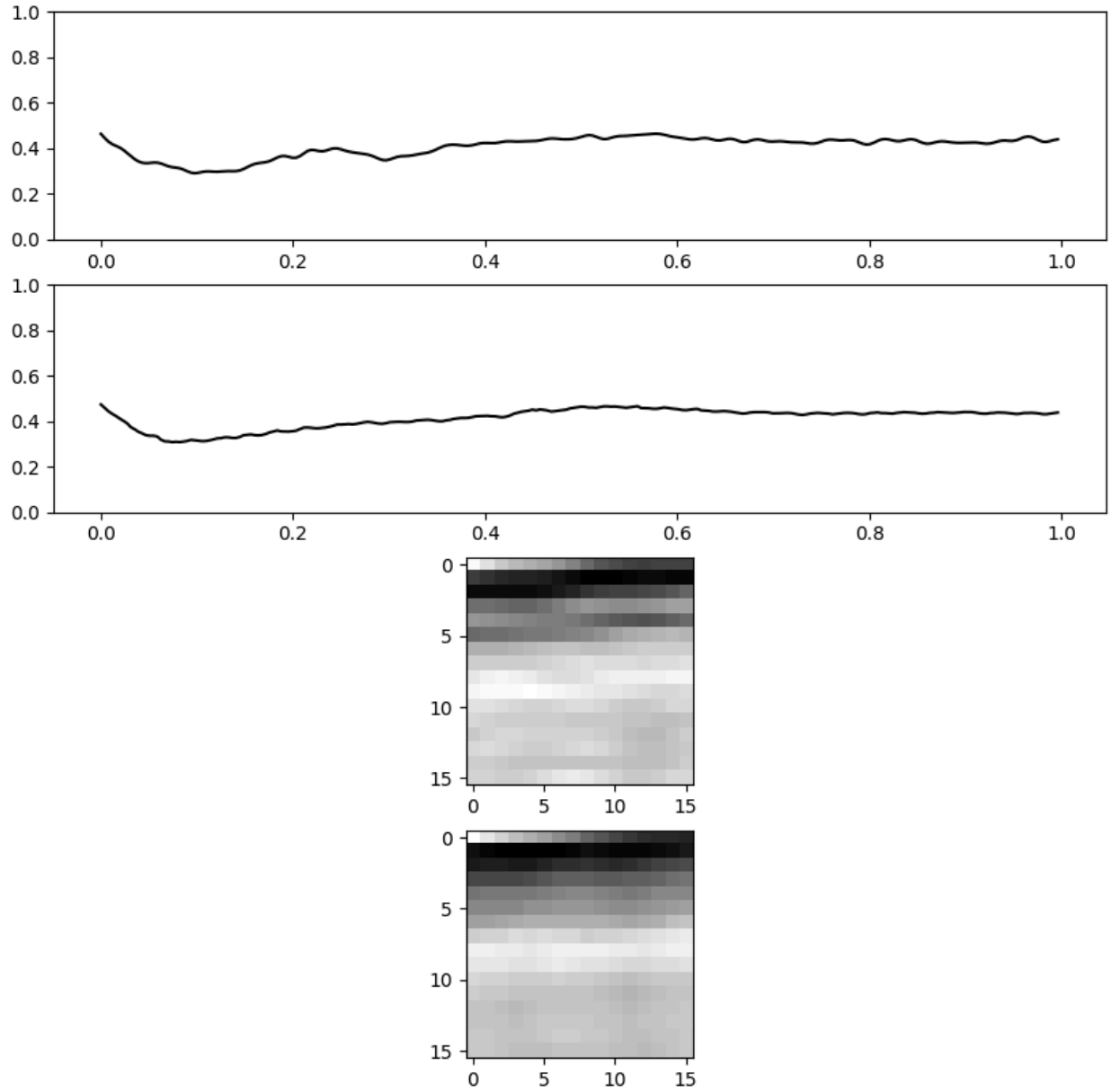
Below are the inputs and outputs of the model, which are 16 x 16 “images,” which is necessary for the convolutional encoding layers to function. The top image is the input passed in, while the bottom image is the reconstructed result after the model encodes and then decodes the input. The reconstructed output, as shown by the images and the line graph, is quite accurate.



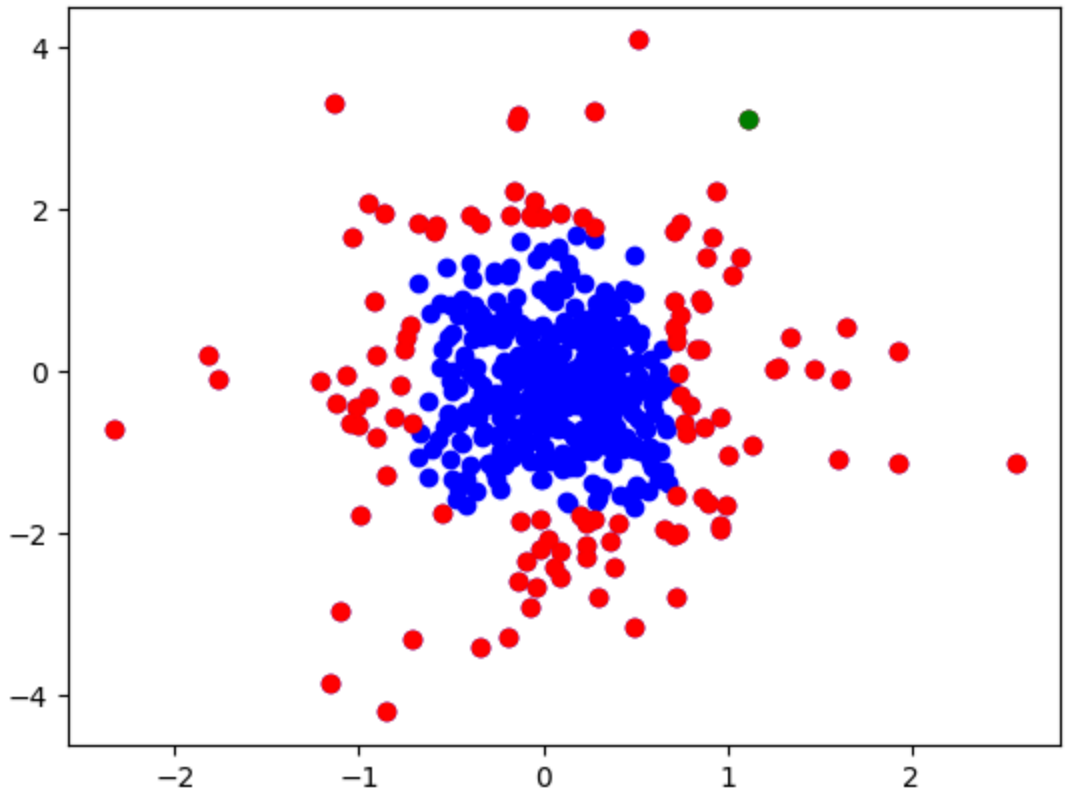




The image above is a scatter plot of 500 encoded inputs. The VAE reduced the inputs to dimension 10, and PCA reduced the inputs to dimension 2. The red dots are items flagged as artifacts, while blue dots are items representing normal activity. The green dot is the input shown above with an artifact, so it is placed among the red dots as it should be.

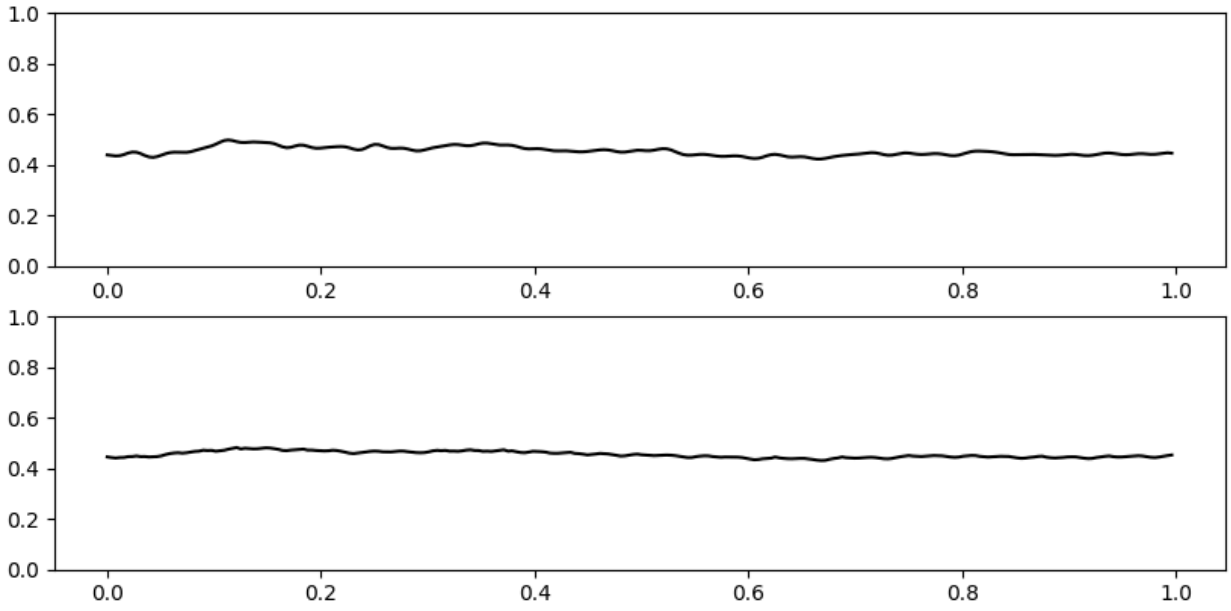


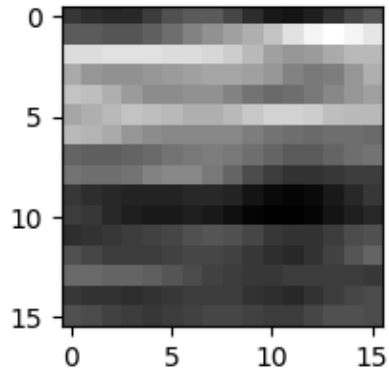
The visualizations above are inputs and outputs of the VAE that similarly contains an artifact. Unlike the other image, there is a downward spike rather than an upward spike.

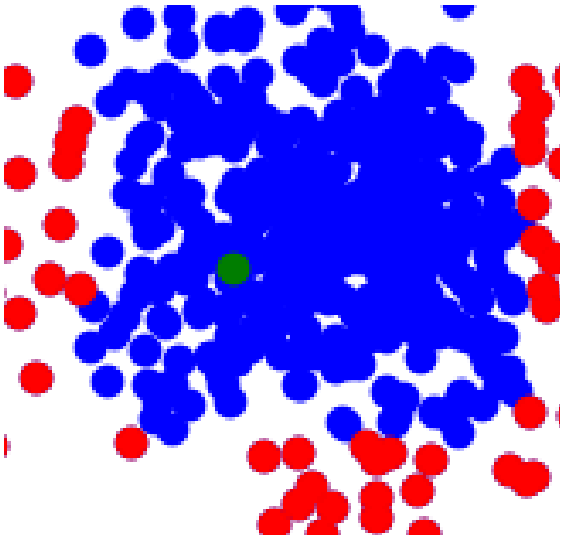


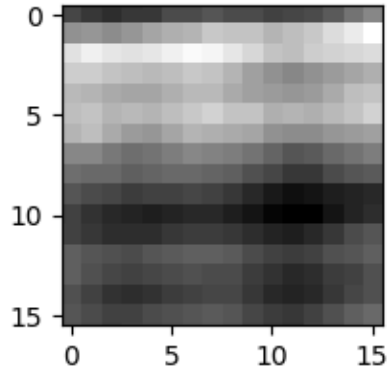
This is a scatterplot with the item shown previously marked in green. The dot is quite far from the first artifact segment shown, which makes sense as it has a downward spike and not an upward spike.

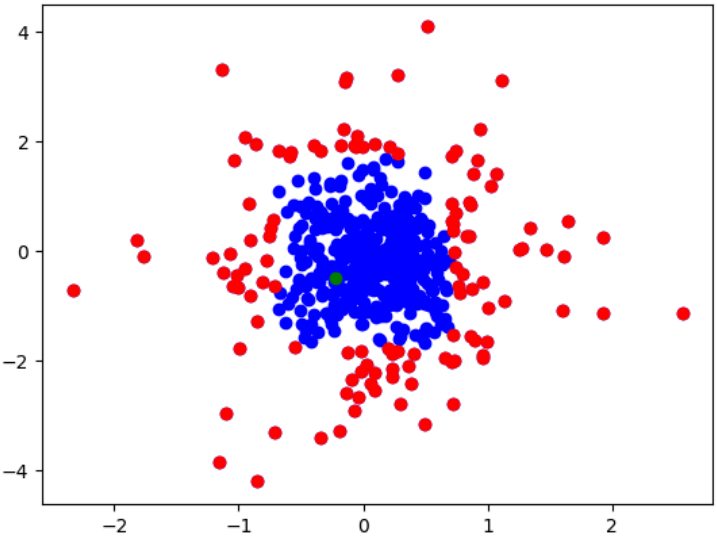
The data below is a segment that is completely normal, with no artifacts.



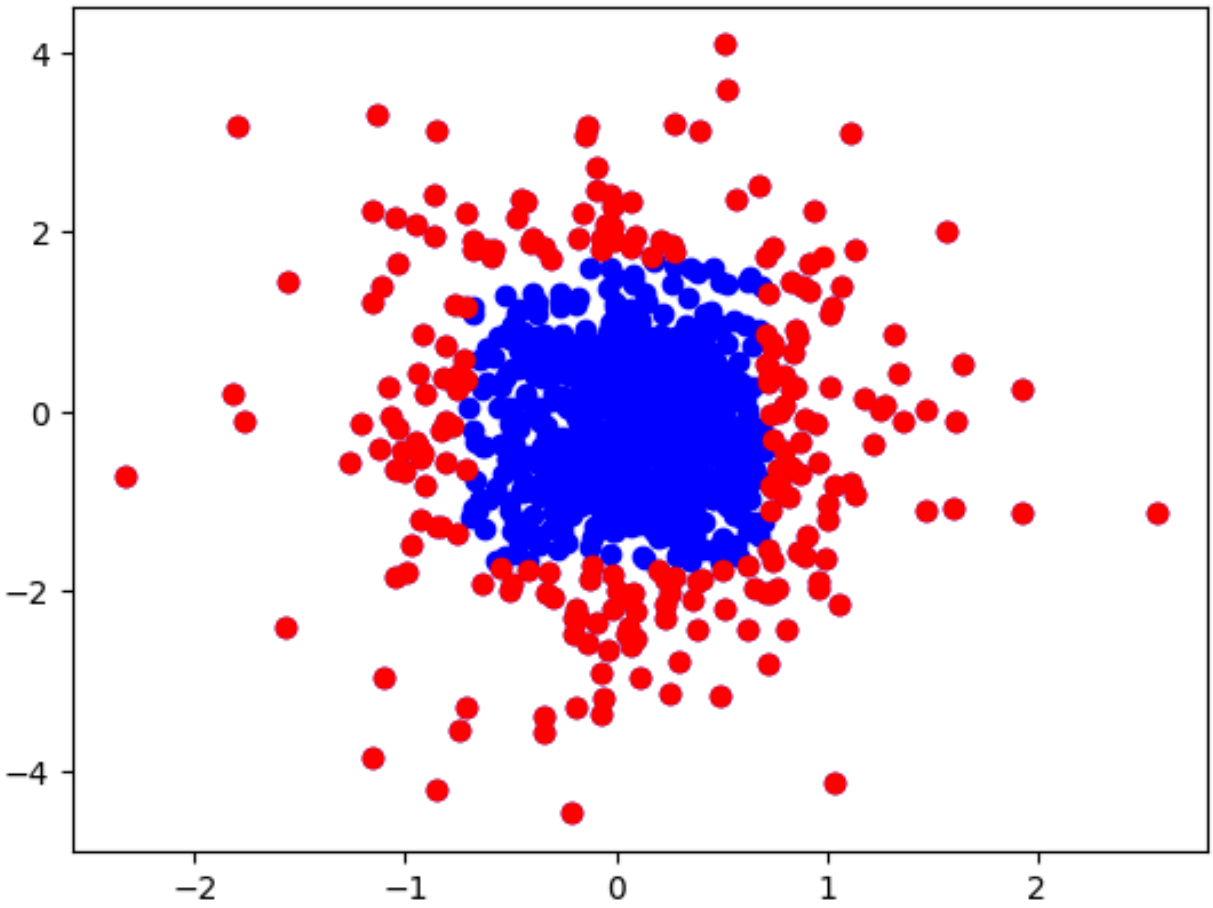






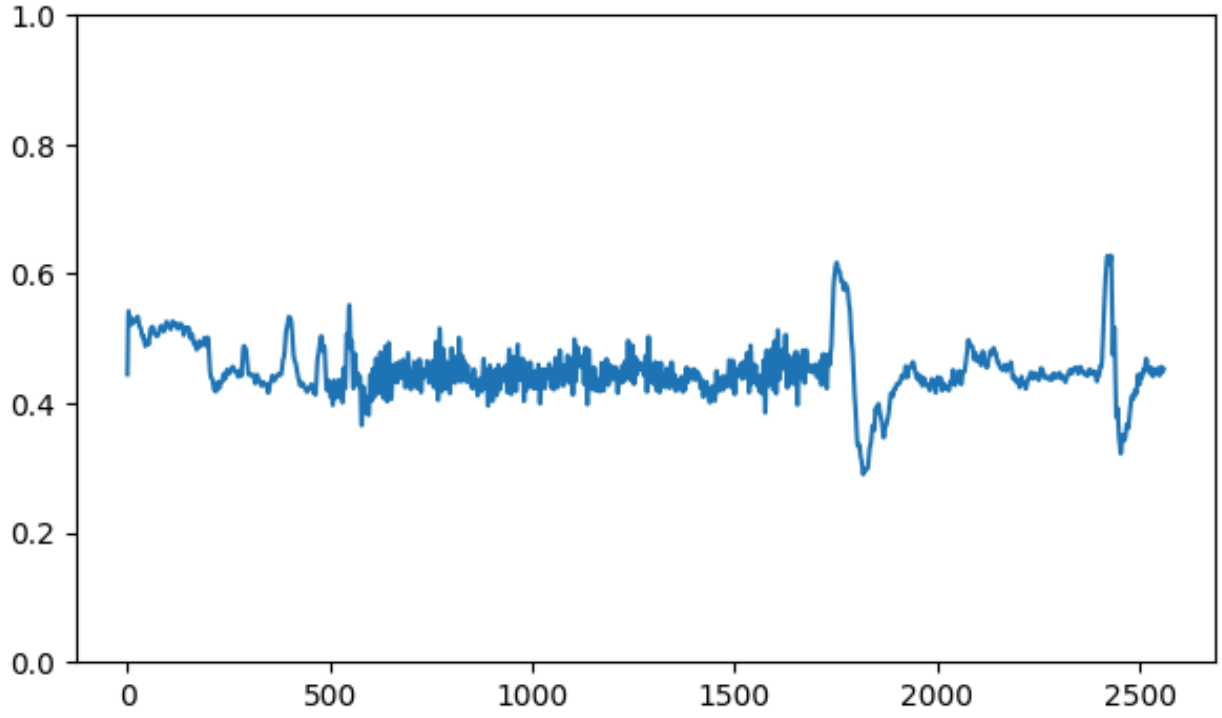


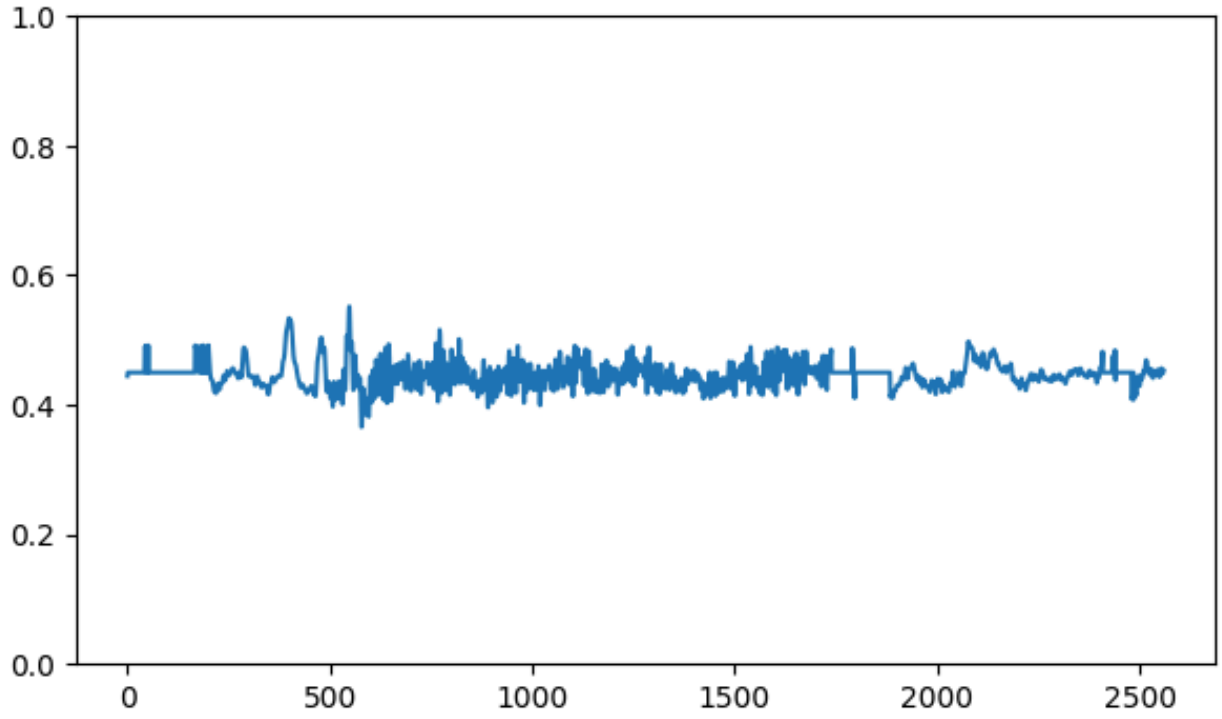
The scattering places the normal input towards the center of the plot.



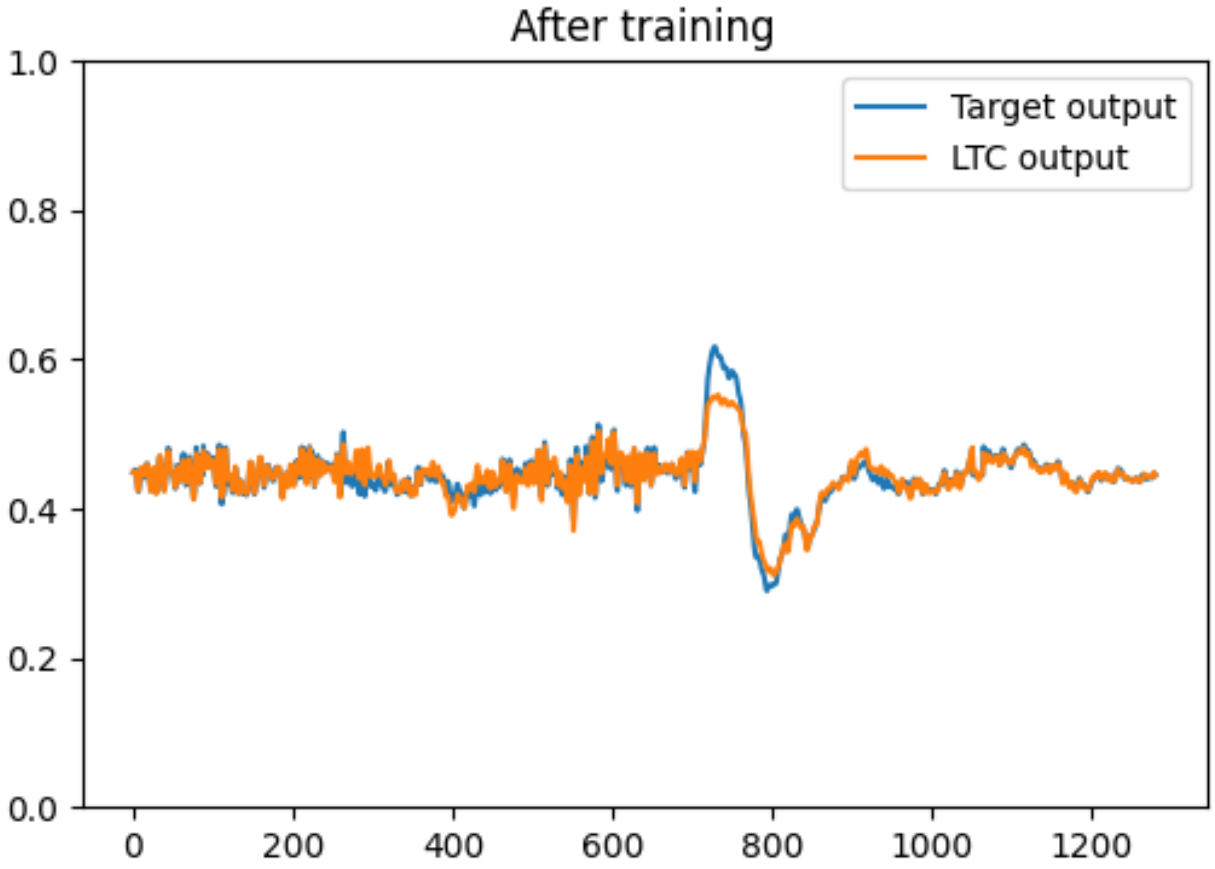
1000 encodings plotted on the same scatter plot.

Images on the next page are results of the artifact removal. The top plot is before the artifacts are detected and removed. The bottom plot is after. All 3 locations of abnormality are correctly removed by the algorithm. The detection algorithm is accurate, but the removal algorithm simply sets data points flagged as artifacts to the mean of the entire data segment. There are better ways of removal, but as the point of the artifact removal is to generate clean data for the LTC, so that the LTC will not accidentally flag artifacts as seizures, this is sufficient. The artifact removal algorithm will not remove seizure data, which tends to last much longer, and the algorithm checks for this.

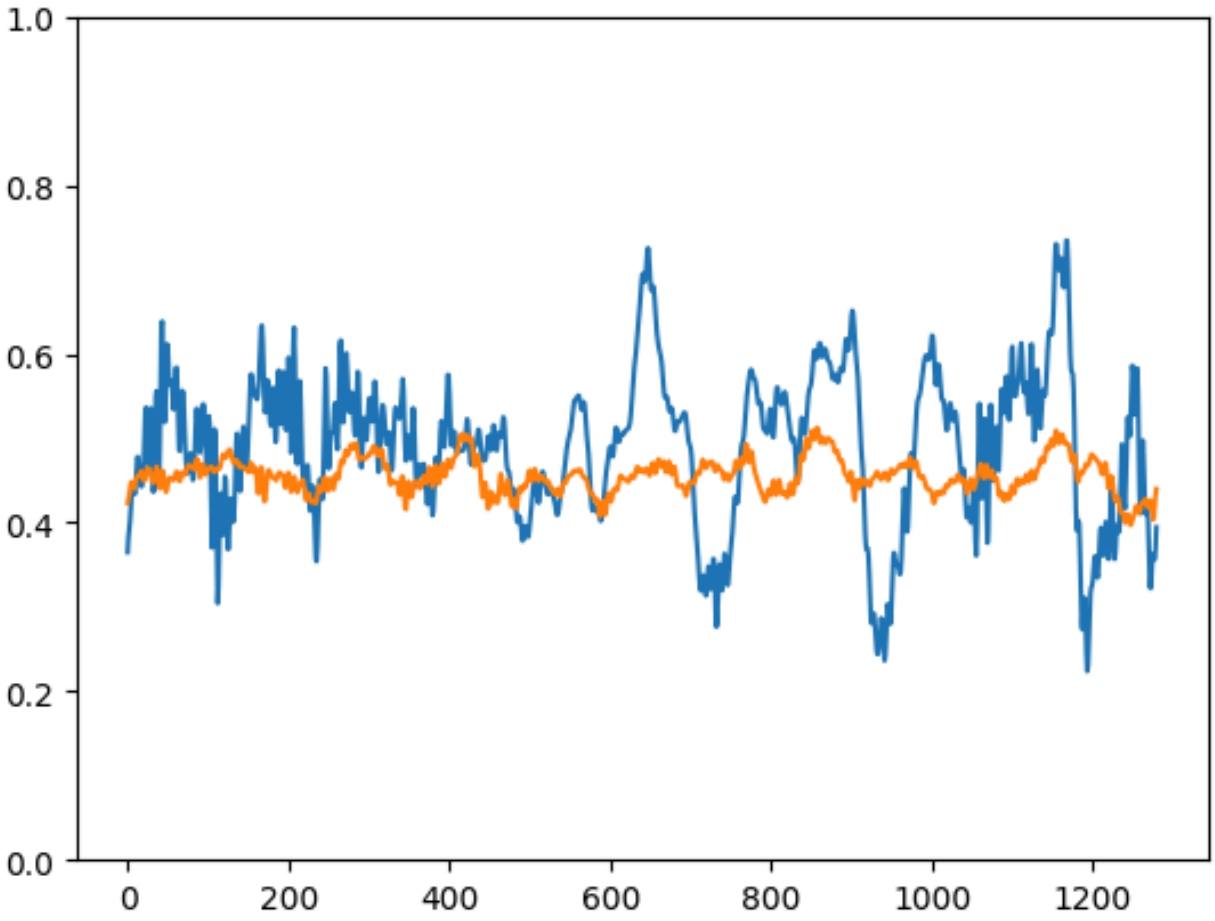

1.0
0.8
0.6
0.4
0.2
0.0
0
500
1000
1500
2000
2500


1.0
0.8
0.6
0.4
0.2
0.0
0
500
1000
1500
2000
2500

Below are results after training of the LTC. The model fits the data well.



The image on the next page is a segment that is part of a seizure. There are way more spikes for a much longer time than would be expected from a segment with only artifacts. The LTC model does a poor job of fitting, as most of the data is normal activity, and so the LTC is better at modeling normal activity.



The large variance between the LTC output and the original data is the reason why the seizure detection algorithm flags this segment as a seizure.

Further research:

Employ a better method of reducing the 10 dimensional data to 2 dimensional data. The strictly linear nature of the PCA algorithm introduced limitations, and was not able to represent the data fully (only about 50% variance explained with first 2 principal components). Potential alternatives include nonlinear methods such as t-SNE and UMAP.
Modify the program to perform seizure prediction instead of just detection. A generative model could be used for this, such as a neural ODE trained as a VAE (Chen et. al, 2019).
Implement a more rigorous and efficient artifact removal algorithm. Due to the goals of the project a less rigorous method was satisfactory, but a faster and more accurate algorithm may increase performance slightly.
This project serves as a validation of the recently developed liquid time-constant network on a database containing real-world data, showing promise for the application of the LTC to other problems. Further research might apply the LTC to other data.