Automatic Seizure Detection Using Lightweight Individual and Ensemble Classifiers

Muhammad Farhan Azmine, Sandra Maria Binoy, Jeremy Decker, and Md Rubel Sarkar

Abstract

Epilepsies are a common, enduring neurological disorder that affects more than 50 million people globally. Clinical electroencephalograms (EEGs) are widely used to diagnose its typically unprovoked, repeated seizures of the same or different forms. High-quality, open-access, and free EEG data can act as a catalyst for ongoing state-of-the-art (SOTA) research works for detection, prediction, and management of epilepsy and seizures using artificial intelligence (AI). When compared to other neurological conditions, epilepsy is less understood since it is characterized by rapid bursts of excessive electricity in the brain that appear as seizures. For the past 30 years, researchers have been striving to predict seizures because they frequently occur without warning. This research relied heavily on EEG since the electrical impulses produced by the brain can be recorded using them. A neurologist often makes the epilepsy diagnosis, which can be challenging in the early stages. Clinicians may be able to diagnose epilepsy and begin treatment sooner if they have para-clinical data to support it from EEG and magnetic resonance imaging. However, due to the necessity for qualified specialists to do the interpretation, EEG capture and interpretation take time and might be expensive. One such approach is the automated detection of correlates of seizure activity that are generalized across several participants and various brain areas. This project explores this concept further and presents a few supervised machine learning approaches that classify seizure and non-seizure records using open source datasets. These approaches include Random Forest, Support Vector Machine (SVM), Artificial Neural Network (ANN), and K-Means. We found that we were able to achieve above an 80% accuracy in single algorithms, even reaching above 90% in singleton training, and that we were able to achieve a highly sensitive network when tested on real time data.

Breakdown of Work

This section breaks down the contribution of each team member to the overall project, covering overall code contributions, where smaller specific functions may be distributed in the comments, as well as work toward the writeup.

Code Breakdown

1. Dataset Segmentation: Main and Real Time Simulation Datasets- Jeremy Decker
2. Feature Extraction
    a. Hjorth Descriptors- Muhammad Farhan Azmine
    b. Time Domain Descriptors- Sandra Maria Binoy
    c. Spectral Power Features- Md Rubel Sarkar
    d. Discrete Wavelet Transform Statistics- Jeremy Decker
    e. Data Harmonization- Md Rubel Sarkar
    f. Anatomically Grouped Features- Jeremy Decker
3. Feature Selections
    a. Information Gain- Md Rubel Sarkar
    b. Mutual Information Feature Selection- Sandra Maria Binoy
    c. Joint Mutual Information- Muhammad Farhan Azmine
    d. Maximum Relevance Minimum Redundancy- Jeremy Decker
4. Model Building, Optimization, Evaluation
    a. K-means- Sandra Maria Binoy
    b. Logistic Regression- Muhammad Farhan Azmine
    c. Decision Trees/Random Forest- Md Rubel Sarkar
    d. Multi-layer Perceptron- Md Rubel Sarkar
    e. Support Vector Machines- Jeremy Decker
    f. Ensemble Methods- Various- All Team members
    g. Real Time Simulation Testing- All Team members

Project Report

1. Abstract- Md Rubel Sarkar
2. Specific Aims- Decided on by all team members
3. Background
    a. Epilepsy, EEG for Seizure Detection- Electrographic Seizure Information- Md Rubel Sarkar
    b. EEG Montages- Jeremy Decker
4. Datasets and Methods-
    a. Dataset Descriptions- Sandra Maria Binoy
    b. Dataset Extraction- Jeremy Decker
    c. Feature Extraction and Algorithms- As in Code Breakdown Item 3, except for Logistic Regression, Written by Sandra Maria Binoy
5. Results and Discussion
    a. As Code Breakdown Item 4
    b. Feature Selection Results- Jeremy Decker
6. Timelines- Jeremy Decker
7. References- All members

1. Specific Aims

The Specific Aims of this project are as follows:

1. **Evaluate the effectiveness of a group of individual classifiers on real-time EEG data.**
2. **Create and evaluate ensemble classifiers based on top performing individual classifiers**
3. **Use additional feature selection techniques to create a lightweight, highly effective classifier.**

2. Background

2.1 Epilepsy

Epilepsy is the abnormal activity of the brain for a certain period due to malfunction of electrical activity. During epilepsy, seizure occurs to the affected person in the form of abnormal behavior, symptoms, sensations or loss of consciousness. It is a burden for almost 1% of the world's population, according to the World Health Organization.[5] The uncertainty of when a seizure occurs greatly affects the quality of life of epilepsy patients. This can lead to anxiety problems even for patients who have infrequent seizures. Patients continue to take medications for avoiding seizures everyday although seizures occur infrequently, which can cause cognitive and physical side effects. To avoid these consequences, the ability to forecast seizures might play a significant role. Also, detection of the type of seizure can pave a new way to make individualized epilepsy treatment possible. For achieving this goal, machine learning techniques are being used on the collected features from the EEG characteristics of the individuals brain signal data.

2.2- Electroencephalography (EEG) for Seizure Detection

There are several types of EEG data that are commonly collected, such as intracranial, scalp, and ambulatory EEG. These signals can be recorded alongside video and images depending upon their use and application.

2.2.1- Electrographic Seizure Information

A large body of research has already been done where different characteristics of certain frequency ranges of the EEG signal have been taken as features, and the seizure types were predicted through machine learning algorithms. There are many types of seizures, such as Focal Non-Specific, General Non-Specific, Tonic-Clonic, Myo-Clonic, Absence, and Partial Seizures. In the past, there have been similar researches using different subjects such as dogs [1] on the basis of intracranial EEG collection. Research on human patients has largely been conducted on scalp EEG data. Different frequency ranges of the EEG data are taken into experiment such as delta (.1-4 hz), theta (4-8hz), alpha (8-12hz), beta (12-30hz), low gamma range (30-70hz). The EEG characteristics of these frequency ranges are taken as features for the machine learning prediction algorithm.

2.2.2- EEG Montages

As a means to collect EEG data in a standard way for both research and clinical applications, the 10-20 System for EEG montages was established. This system standardizes electrode spacing to account for variation in head size, and ensures electrodes will be evenly spaced regardless of head size, enabling researchers to collect data from consistent parts of the brain. All major scalp-accessible brain regions are covered, tracking frontal, parietal, temporal, central, and occipital regions of the brain. Additionally, the numbering system makes it easy to separate the hemispheres of the brain, with even numbers on the right side, odd on the left, and Z indicating the presence of an electrode on the midline.[15] This naming convention is used with many EEG studies, polysomnography, and portions of it are even used in intracranial recordings, although intracranial localization is significantly more complex. Figure 1 below shows an example of a 16-channel standard montage using the 10-20 System.

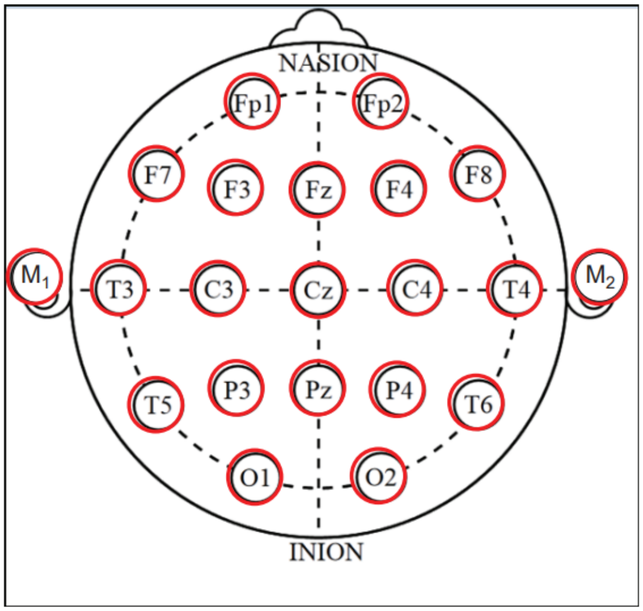

Figure 1. An example of a 16-electrode EEG setup using the 10-20 System. M electrodes represent electrodes placed on the mastoid bones on the skull. The inion and nasion are bones commonly used in alignment measurements for this system.[15]

3. Datasets and Methods

3.1- Datasets

For this investigation, two EEG datasets were chosen based on their availability, continuous data type, and relatively standard montages. Each dataset is described in detail below:

3.1.1- CHB-MIT Scalp EEG Database:

This database is a pediatric database involving 22 subjects with intractable, or drug resistant, epilepsy. It contains continuous EEG data from these subjects, typically recording between 2 and 4 hours, unless a subject had a seizure, in which case the file size is less. This dataset provides two unique facets; it contains continuous data from these subjects, allowing for better real-time detection testing, and it works with pediatric epilepsy, which may have different facets than adolescent and adult epilepsy. Seizures are labeled within the dataset, making class labeling easy for machine learning applications. Additionally, montage and other relevant information is readily available, allowing researchers to easily align channel inputs, and group signals based on location. All data in this dataset are freely available and open source.

3.1.2- Siena Scalp EEG Database

The database consists of EEG recordings of 14 patients and subjects include 9 males (ages 25-71) and 5 females (ages 20-58).They were monitored with a Video-EEG at a sampling rate of 512 Hz, with electrodes arranged on the basis of the international 10-20 System. This dataset provides insights to adult epilepsies and the seizures are labeled within the dataset. This again makes differentiating between seizures and non-seizures easy for machine learning applications. It also contains Non-invasive EEG data that is useful for exploring the possibility of developing a noninvasive monitoring/control device for the prediction of seizures for researchers. They are generally noisier than those obtained with other invasive (intracranial) techniques.However, the adoption of noninvasive data collecting techniques would enable the procedure to be used in real-time in portable and wearable devices with little discomfort for the patients. The data is freely available and open source.

3.2- Methods and Experiments

3.2.1- Overview of Experimental Methods

The following section describes the methods used to extract the datasets used in this process, the feature extraction process, and the algorithms used and optimized in this project. Figure 2 below describes the basic process of the data processing pipeline. The data started as full, long form EEG recordings, where data segmentation techniques extracted all seizure data in 5-second segments, and an approximately equal amount of non-seizure segments in order to create a well balanced dataset. From there, the data was passed through a series of feature extraction techniques to quantify a number of important electrographic characteristics of the signal in both the time and frequency domain to distinguish between seizures and non-seizures, as it is well proven that these differences exist. [] As a newer portion of this experiment, we investigated whether or not anatomical grouping would be able to create more informative features than individual channels. To do this, we utilized the mapping system used in the 10-20 system, with slight modifications for the bipolar montage, where two channels are subtracted to form a single signal, to group channels of EEG data, and extracted new grouped features for each previously tracked feature. Finally, before training and evaluating classifiers, we utilized a variety of feature selection techniques to shrink the feature space.

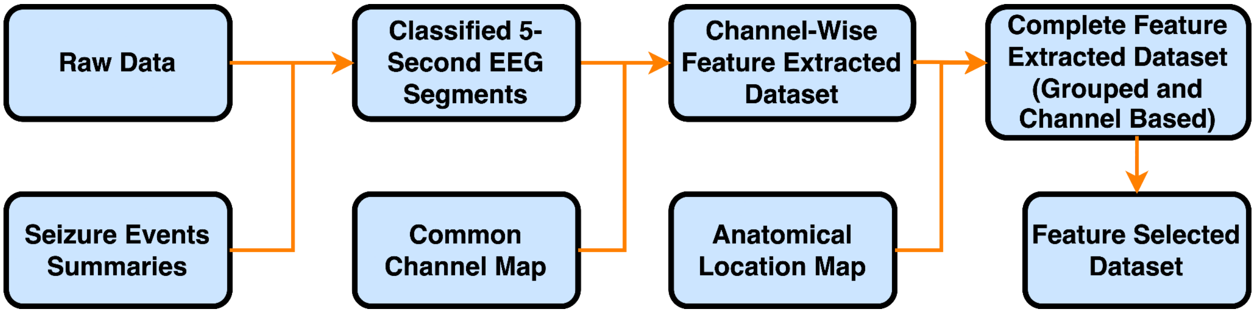
Raw Data
Classified 5-Second EEG Segments
Channel-Wise Feature Extracted Dataset
Complete Feature Extracted Dataset (Grouped and Channel Based)
Seizure Events Summaries
Common Channel Map
Anatomical Location Map
Feature Selected Dataset

Figure 2. Data Processing Flow for the MIT and Siena Databases, resulting in a feature selection database ready for use in a classifier.

Training and evaluating classifiers took place in three phases. In this project, we investigated a variety of lightweight classifiers, including the decision tree and random forest, support vector machine, logistic regression and K-means clustering. The first phase of model evaluation was to use k-folds to split the dataset for training and testing, with the exception of its use for K-means, which requires a larger training set in order to generalize well. After optimizing using a variety of selected feature spaces and model parameters, we established the most effective models. In the second portion of evaluation, these top ranking models were combined in different ways to form ensemble classifiers, which were evaluated in a similar fashion. Finally, we utilized a new dataset segmented in a similar way to the main evaluation set, differing by instead segmenting each EEG file into 5 second segments and labeling them based on the presence of seizures. This dataset was used only for testing pre-trained individual and ensemble models, in order to provide a unique look into the generalizability of each.

3.2.2- Dataset Segmentation

Both the MIT and Siena databases had text files summarizing the file location and times of seizures for each subject. Text scrapers written in MATLAB were utilized to identify these times and reshape the data into a usable format for segmentation.

To create the dataset used for the first two phases of model evaluation, seizure segments were extracted by breaking each seizure into 5-second chunks, and saving them into files for feature extraction. Non-seizure segments were collected by taking a number of randomly sampled 5-second segments that did not start within 10 seconds of a seizure and saving them, with elimination.

Differing from the original protocol, the segmentation script for the real time simulation dataset instead took 5 second segments starting from the start time of the file continuing until the file's end, labeling each based on whether or not a seizure was at least partly contained between its start and end.

3.2.3- Feature Extraction

To extract useful information from the raw EEG signal, feature extraction techniques in both the time and frequency domains will be used. Additionally, neuroanatomy will be used to extract additional features.

In the time domain, we will extract average peak amplitude, the signal's zero crossing rate, and its Hjorth descriptors. The amplitude of the EEG pattern represents the pattern's intensity in microvolts of electrical energy. In general, when frequency lowers, the amplitude of the EEG increases. The amplitude of an electroencephalogram (EEG) wave is the measure of voltage across the brain at any given time.High-amplitude waves indicate that many neurons are firing together. Thus amplitude is an important feature for classification of seizures. So the average peak amplitude from the absolute value of signal in each channel is calculated and extracted as a feature. Zero crossing is a point where the amplitude drops from positive to negative or vice versa. Zero crossing rate will be found by tracking successive sign changes. These zero-crossing patterns can extract the spatiotemporal structure of scalp voltage fluctuations and allow detection of these intracranial inter ictal discharges that are subtle, low-amplitude waveforms below scalp EEG background which otherwise is not detected. Thus it is important for epilepsy seizure detection. Interspike interval is calculated by finding the difference between spike amplitude and then calculating the median of those differences. These features are important in predicting interictal spikes which helps in classifying epilepsies

Peak amplitude will be calculated from the absolute value of the signal and the zero crossing rate will be found by tracking successive sign changes. The Hjorth descriptor was chosen due to its use in previous seizure detection work, returning three descriptors; Activity, calculated from the variance of the signal to establish signal power; Mobility, calculated using the square root of the ratio between the variance in the signal and its first derivative, finding the mean frequency of the power spectrum; and Complexity, which tracks the change in frequency, calculated using the ratio of the mobility of the signal's derivative and the original signal [5]. Equations 1, 2, and 3 of Table 1 describe how each function is calculated.

In the frequency domain, we will use the Fast Fourier Transform(FFT) and discrete wavelet transform(DWT) to extract features. We will observe changes in the whole spectrum, as well as specific changes in the delta, theta, alpha, beta and gamma bands, as they have significance neurologically. The FFT is a faster form of the DFT, which estimates the spectral content of a signal using Equation 4 in Table 1 below. The DWT works differently, as it decomposes the signal into wavelets defined only for a short time, based on transforms of a common mother wavelet. This method is particularly useful for non-stationary signals, which makes it an operation of interest in tracking fast-changing neural dynamics like seizures. It is calculated by applying a high and low pass filter and downsampling the data, isolating the high and low frequency components of the signal. This process is repeated with the low pass filtered data until the frequency resolution is sufficient for the target application. A one-level DWT like the one used in this investigation will produce two vectors: approximation coefficients, or low frequency content, and detail coefficients, or high frequency content. [6] In this investigation, we will take the variance, root mean square, and entropy of these measures and utilize them as features.

In addition to extracting features from the raw signals themselves, we will apply basic neuroanatomy to track more local changes in these features. We plan to extract the time based features, as well as the power spectrums, from the averaged signals from the frontal, central, and occipital regions of the brain, based on electrode locations using the 10-20 system.

Equation	**Key Variables**
1. $var(y(t))$ = Activity	y(t) - Original Signal
2. $\sqrt{\frac{var(\frac{dy(t)}{dt})}{var(y(t))}}$= Mobility; 3. $\frac{Mobility(\frac{dy(t)}{dt})}{Mobility(y(t))}$ = Complexity	$\frac{dy(t)}{dt}$- First derivative of the signal
4. $X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}$ where k = 0, …, N-1	X_k = frequency domain value of the kth spectral line of series x_n. N- number of entries in x_n n- numbered entry in x_n k- DFT coefficient number, or spectral line

Table 1: Table of Feature Extraction Equations

3.2.4- Feature Selection Techniques

Feature selection is a critical part of model building in classification models. In this project, we will utilize Information Gain(IG), Joint Mutual Information (JMI), and Maximum Relevance Minimum Redundancy (MRMR). The rankings from these techniques will be used to optimize network performance and create a lightweight, high performing network.

Information Gain tracks the amount of uncertainty that can be eliminated after utilizing the variable in question. To do this, the entropy, or uncertainty, of the class variable is calculated, and is subtracted by the remainder of entropy after a feature is applied, giving an amount of information gained by utilizing the feature. Equations 1, 2, and 3 in Table 2 describe calculating entropy, the remainder, and information gain. JMI and MRMR utilize Mutual Information, which is built off of conditional entropy, shown in Equations 4 and 5 below. Mutual information based feature selection builds upon base mutual information by subtracting the mutual information between a feature and all other features in a dataset from the calculation between a variable and its target. It uses a weight to determine how much this measure should count, with 0 being pure mutual information maximization. JMI identifies the features that have the most complimentary information, or information shared with other features, in the dataset, calculated using the sum of the entropy of joint random variables representing the combinations of two features as compared to the class labels. In contrast, MRMR seeks to reduce the amount of redundant information by subtracting the mutual information of a feature and the class label by the average of the mutual information of the target feature and all other features. JMI and MRMR are defined by Equations 6 and 7 in Table 2, respectively. [7]

Equation	**Key Variables**
1. $H(X) \ =- \sum_{x\varepsilon X} p(x)log(p(x))$	H(X)- Entropy of variable X x= possible value of x
2. $rem(d,X) \ =- \sum_{l\varepsilon levels(d)} \frac{\lvert D_{d=l}\rvert}{\lvert D\rvert} H(Y_{d=l})$	d - A feature in the dataset Y- Class label variable $H(Y_{d=l})$- Entropy of the subset of Y where d= level l
3. $IG(d,Y) \ = H(Y) \ - \ rem(d,Y)$	d= A feature of the dataset

4. $H(X\lvert Y) \ =- \sum_{y\varepsilon Y} p(y) \sum_{x\varepsilon X} p(x\lvert y)log(p(x\lvert y)\ )$	H(X|Y)-Conditional entropy of X given Y
	y= possible value of Y
	x= possible value of X

5. $I(X;Y) \ = \ H(X) \ - \ H(X\lvert Y)$	I(X;Y)- Mutual Information of X and Y
6. $J_{jmi}(X_k) \ = \ \sum_{X_j\varepsilon S} I(X_kX_j;Y)$	$J_{jmi}(X_k)$- Joint Mutual information for variable X_k X_k and X_j- Variables in dataset S
7. $J_{mRMR}(X_k) \ = \ \sum_{X_j\varepsilon S} I(X_kX_j;Y)$	X_kX_j- Joint random variable of two features in dataset S
8. $J_{mifs}(X_k) = I(X_k;Y) - \beta \sum_{X_j \in S} I(X_k;X_j),$	B- weighting factor for the summation.

Table 2: Table of Feature Selection Techniques

3.2.5- Algorithms

3.2.5.1- Random Forest Classifier:

A random forest is a machine learning method for classification and regression problems. It makes use of ensemble learning, a method for solving complicated issues by combining a number of classifiers. In a random forest algorithm, there are many different decision trees. The random forest algorithm creates a "forest" that is trained via bagging or bootstrap aggregation. The accuracy of machine learning algorithms is increased by bagging an ensemble meta-algorithm. The accuracy of the result grows as the number of trees increases. The decision tree algorithm's shortcomings are eliminated with a random forest. It improves precision and lowers dataset overfitting.

For classification tasks Gini Index and Entropy are used to decide how nodes on a decision tree branch[8].

$$Gini \ = \ 1 \ - \ \sum_{i=1}^{C} p_i^2 \ , Entropy \ = \ \sum_{i=1}^{C} - \ p_i \ log_2 p_i$$

This formula calculates the Gini of each branch on a node based on the class and probability, indicating which branch is more likely to occur. In this case, pi stands for the class's relative frequency in the dataset, and c for the total number of classes. Entropy is more mathematically complex than the Gini index because it is calculated using a logarithmic function.

3.2.5.2- Support Vector Machine (SVM):

In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. SVMs are one of the most robust prediction methods, being based on statistical learning frameworks[9]. SVM works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found, then the data are transformed in such a way that the separator could be drawn as a hyperplane[10]. Following this, characteristics of new data can be used to predict the group to which a new record should belong.

We define the hypothesis function h as:

$h(x_i)=\{+1-\ \text{if } w \cdot x + b \geq 0,\ -1\quad if\ w{\cdot}x\ +\ b\ <\ 0$

The point below the hyperplane will be assigned a class ranking of -1, while the point above or on the hyperplane will be assigned a class ranking of +1. Consequently, the SVM learning algorithm's main objective is to identify a hyperplane that can accurately separate the data. These hyperplanes could be numerous. The best one, also known as the ideal hyperplane, must be identified.

3.2.5.3- Artificial Neural Networks(ANN):

The design of an artificial intelligence neural network replicates a brain network in theory. Artificial neurons are made up of a neuron body and connections to other neurons since natural neurons are composed of layers of neurons that are directionally coupled. Layers of neurons are added to create a neural network. It has the ability to learn, recall and generalize from the given data by suitable assignment and adjustment of weights. They have the ability to work with incomplete data and thus good for real time seizure detection.

The input layer houses the nodes that take in information from datasets . The bottom layer of a neural network is referred to as the output layer, which makes a prediction based on the processed input from the rest of the network. Between the input and output layers, there may be one or more hidden layers. The majority of calculations are carried out by hidden-layer neurons during the function's approximation. In a fully connected or dense neural network, each neuron in a particular layer is connected to all neurons in the previous and next layers. Each of these connections carry a weight, noted by two indices. The first index is the receiving neuron number, and the second index is the sending neuron number. Each network layer is assigned a bias. The bias is similar to the weight assigned to a neuron, but applied to the entire layer. When the network processing starts, the initial values for weights and biases are usually randomly set. In order to calculate the output from each node in a layer the weighted sum of the inputs are passed into an activation function. This activation function can be sigmoid, tanh, linear etc. The function then determines the output values to the nodes in the next layer. This network processing can be done in two ways : forward propagation which is basically calculating from the input layer to the output layer , and backward propagation in which the calculation is done from the output layer to the input layer. Back propagation uses error functions to calculate discrepancies between the predicted output and resulting output in order to adjust the weights of the network to minimize error. Due to this error minimization we use backpropagation for classification of seizures.

3.2.5.4- Clustering (K-means):

Clustering is the task of categorizing a population or set of data points so that data points in the same group are more similar to data points in other groups than data points in other groups. Simply put, the goal is to separate groups with similar characteristics and place them in clusters. Clustering is critical because it determines the intrinsic grouping of the unlabeled data. The K-means clustering algorithm is the most basic unsupervised learning algorithm for clustering problems. The K-means algorithm divides n observations into k clusters, with each observation belonging to a cluster prototyped by the nearest mean.[13] The algorithm iteratively minimizes the distances between every data point and its centroid in order to find the most optimal solution for all the data points. The objective function for the K-means clustering algorithm is the squared error function:

$$\Sigma_{i=1}^{k}\Sigma_{j=1}^{n}(||x_i - v_j||)^2 = 1 \qquad [15]$$

The advantage of using k-means is that it is simple to implement and scales to larger data sets, but clustering outliers can be challenging. So the outliers should be removed or clipped before clustering.[14]

3.2.5.6- Logistic Regression:

Logistic regression is a supervised classification algorithm at its core. For a given set of features (or inputs), X, the target variable (or output), y, can only take discrete values in a classification problem.
The model constructs a regression model to predict the likelihood that a given data entry belongs to the category labeled "1." Logistic regression models the data using the sigmoid function, just as linear regression assumes the data follows a linear function. Only when a decision threshold is introduced into the equation does logistic regression become a classification technique.[16] The threshold value is an important aspect of Logistic regression and is determined by the classification problem itself. To build a logistic regression model,we simply pass the output of the basic linear regression model through the logistic function.

$$\mathrm{M(d)} = 1/(1 + e^{-w^*d})$$

Before we train a logistic regression model we map the binary target feature levels to 0 or 1.[17]
Unlike decision trees or support vector machines, this algorithm allows models to be easily updated
to reflect new data which is good for us when we simulate the model for real time datasets.
Stochastic gradient descent can be used to update the model.

3.2.6- Experimental Methods

In order to evaluate the objectives of our aims, we will utilize two separate subsets of both the Siena and the MIT dataset. The first subset of the data will consist of the bulk of the seizure data, and an even amount of non-seizure data from each dataset. This subset will be used in the main evaluation of all Aims, but in particular Aims 1 and 2. The second subset of the data will simulate a real time dataset by maintaining the balance that would be seen in a real-world scenario, by creating samples from entire EEG session files, regardless of dataset balance. Features from the dataset will be extracted using the techniques described in Section 3.2.4 of this report, and feature selection methods will be utilized to identify important features to be considered during the optimization of each classifier.

To fulfill Aim 1, each classifier mentioned in section 3.2.5 will be trained on the subsets of the feature space based on feature selection techniques. K-fold cross validation will be used to avoid bias, and accuracy and the confusion matrix will be utilized as metrics for comparison. The results of this optimization will be used to select the top algorithms for the combinative models. Special attention will be given to classifiers with particularly high sensitivity, as always identifying seizures is critical for the application this algorithm would ideally be used for. We expect to achieve between 80 and 90% accuracy overall in this phase.

To fulfill Aim 2, combinations of top performing classifiers will be utilized to create an ensemble classifier either by using a decision tree to add weights to the decision, or simple voting methods. These ensemble methods will be trained on the first subset of the data and evaluated for performance. A few top performers will be chosen and their trained versions will be evaluated on the second subset of the data. We expect the ensemble algorithms to exceed 90% accuracy overall in the first subset, and have a high sensitivity in the second while maintaining high accuracy.

4. Results and Discussion

4.1- Feature Selection Results

We utilized a variety of feature selection techniques to find important features in the datasets. The top 10 features of the MIT and the Siena Database are shown in Tables 1 and 2 below. In analyzing these features, we find a few patterns amongst the dataset that were particularly of note. First, there is an extremely strong presence of temporally located features in the dataset, indicating that this region is especially important in detecting and determining seizures. Temporal features hold the top ranked variable in every method in both datasets save Mutual Information Based Feature Selection, where an Occipitally based zero crossing rate was much higher ranked. In talking with Dr. Sujith Vijayan, a professor in the neuroscience department, we found that the Temporal Region of the brain is highly correlated with seizures, further highlighting the importance of this region. Frontal regions also had a large presence in the selected features, indicating that it may also be highly correlated with seizures. Many of the frontal electrodes selected also had coverage in or near temporal areas, which may play a role in their importance.

In addition, features related to the low frequency composition of the EEG data were also highly represented in all feature selected algorithms, with many lists ranking factors of the DWT Approximation coefficients, Theta Power, and Delta Power appearing in many different rankings. In particular, the variance of the Approximation coefficients appeared to play a role, and this is further supported by the fact that the complexity of the overall signal is also a common signal. This indicates that a shift in the rate of change of the signal can indicate the onset of a seizure, something that can actually be seen on EEG in some rare cases, as the power of most waves drops just before a seizure. To further support the changes in the frequency content of the signal, the interpeak intervals and zero crossing rates of the signal also were found to be important signals, and can easily show this type of change.

In conclusion, we found that temporally located features play a critical role in determining the presence of a seizure. In addition, features that investigate low frequency content or affect it can be useful in this measurement. These findings could indicate that a system focused on the Temporal region of the brain could be used as a more lightweight setup for seizure monitoring when localization is not an issue.

Table 1: Features for MIT Dataset:

Method /Rank	Information Gain	Mutual Information Based Feature Selection	Joint Mutual Information	Maximum Relevance Minimum Redundancy
1	T7-P7 Variance	P8-O2 Zero-Crossing Rate	T7-P7 Theta Power	F7-T7 DWT Approximation Coefficient RMS
2	T7-P7 DWT Approximation Coefficient Variance	FP1-F7 Complexity	T7-P7 DWT Approximation Coefficient Variance	F3-C3 Zero Crossing Rate
3	P7-O1 DWT Approximation Coefficient Variance	F4-C4 Zero-Crossing Rate	Frontal-Temporal Median Variance	Cz-Pz Inter-Peak Intervals
4	F7-T7 DWT Approximation Coefficient RMS	F3-C3 Inter-Peak Interval	T7-P7 Alpha Power	F7-T7 Theta Power
5	T7-P7 DWT Approximation Coefficient RMS	F8-T8 Zero-Crossing Rate	Temporal-Parietal Median DWT Approximation Coefficient Variance	Temporal-Parietal Median DWT Approximation Coefficient RMS
6	F3-C3 Inter-Peak Interval	P8-O2 DWT Detail Coefficient Variance	Temporal-Parietal Median Theta Power	T7-P7 Delta Power
7	C3-P3 Inter-Peak Interval	Fp1-F3 DWT Detail Coefficient Entropy	T7-P7 Variance	P7-O1 Theta Power
8	P3-O1 Inter-Peak Interval	F8-T8 DWT Detail Coefficient Entropy	Parietal-Occipital Median Theta Power	Temporal-Parietal Mean DWT Approximation Coefficient RMS
9	CZ-PZ Inter-Peak Interval	T8-P8 Complexity	Temporal-Parietal Mean DWT Approximation Coefficient Variance	Temporal-Parietal Median Theta Power
10	F7-T7 Theta Power	F3-C3 DWT Detail Coefficient Entropy	P7-O1 Theta Power	Frontal-Temporal Median DWT Approximation Coefficient RMS

Table 2: Features for Siena Dataset:

Method/ Rank	Information Gain	Mutual Information Based Feature Selection	Joint Mutual Information	Maximum Relevance Minimum Redundancy
1	T4 Variance	T3 Gamma Power	Temporal Median Theta Power	T4 Theta Power
2	F10 Variance	Fp1 DWT Detail Coefficient Entropy	Temporal Median Variance	Fc1 DWT Approximation Coefficient RMS
3	O1 Inter-Peak Interval	Fc6 Alpha Power	T4 Theta Power	F3 DWT Detail Coefficient Entropy
4	T3 Inter-Peak Interval	T4 DWT Detail Coefficient Entropy	T3 Delta Power	O2 Mobility
5	T5 Inter-Peak Interval	F9 DWT Approximation Coefficient Entropy	Temporal DWT Median Approximation Coefficient Variance	F8 Peak Average Amplitude
6	P4 Inter-Peak Interval	T5 Delta Power	Temporal Mean Variance	T5 DWT Detail Coefficient Entropy
7	T6 Inter-Peak Interval	F9 Alpha Power	Frontal-Central Median Theta Power	Frontal Median Theta Power
8	EKG 2 Inter-Peak Interval	C4 DWT Detail Coefficient Entropy	T5 Delta Power	Occipital Median DWT Approximation Coefficient RMS
9	T5 DWT Approximation Coefficient Variance	F10 Zero Crossing Rate	T5 DWT Approximation Coefficient Variance	EKG 2 Alpha Power
10	T4 DWT Approximation Coefficient Variance	Occipital Mean Inter-Peak Interval	F8 Theta Power	Frontal Mean DWT Detail Coefficient Entropy

4.2- Model Results

4.2.1 Random Forest Classifier (RFC):

Before running RFC Information Gain (IG) feature selection technique applied to extract top 2%, 5% and 10% features from the processed MIT and Siena Database. With the selected feature RFC algorithm has been run for different max_depth parameters. They achieved the following accuracy result shown as a table.

To get the best parameter values hyper parameter tuning has been done for both MIT and Siena databases. Parameter has been tuned for criterion, max depth, n_estimator and min_samples_leaf variables.

Tuned Parameters:

Best Estimator : criterion = entropy, max_depth = 15, min_samples_leaf = 5, n_estimators = 100
With these tuned parameter classifiers have been run. Results for different datasets have been recorded in the results and discussion section.

Table: MIT and Siena Dataset Accuracy with top 2%, 5%, 10% selected features.

Classifier=Random Forest, estimator =100, criterion=entropy	Dataset	Accuracy (%) With Top 2% Features	Accuracy (%) With Top 5% Features	Accuracy (%) With Top 10% Features
Max_depth = 5	MIT	85.02	85.02	84.92
	Siena	81.34	89.50	89.49
Max_depth = 10	MIT	85.20	85.95	87.17
	Siena	82.81	89.78	89.77
Max_depth = 15	MIT	85.11	86.71	87.64
	Siena	83.68	89.78	89.77
Max_depth = 20	MIT	85.39	86.71	87.36
	Siena	83.68	89.78	89.77

From the above table it can be seen that accuracy is higher when the model gets top 10% of the features. It decreases for top 5% features and 2% slightly. Overall we get best accuracy for the top 5% features.

Table: Real Time Dataset MIT and Siena Dataset Accuracy with top 5% selected features.

Classifier = Random Forest	Dataset	Confusion Matrix	Accuracy (%) With Top 5% Features

Criterion = Entropy,
Max Depth =10,
Estimator =100
	MIT	[[2538 3]
		[8 36]]
			99.49

	Siena	[[2131 3] [10 21]]	99.35

4.2.2 Multi Layer Perceptron (MLP) Classifier:

Top 2%, 5% and 10% features have been used as in DTC. The achieved result has been shown in the table below. Hyperparameter tuning has been done. Found the best parameters as below.

Tuned Parameters:
Solver =adam, activation =relu, learning rate= constant.

Table: MIT and Siena Dataset Accuracy with 2%, 5%, 10% selected features.

Classifier	Dataset	Accuracy (%) With Top 2% Features	Accuracy (%) With Top 5% Features	Accuracy (%) With Top 10% Features
MLP Classifier	MIT	82.96	84.46	85.39
	Siena	67.35	86.59	65.01

Table: Real Time MIT and Siena Dataset Accuracy with top 5% selected features.

Classifier	Dataset	Confusion Matrix	Accuracy (%) With Top 5% Features
MLP Classifier	MIT	[[2539 2] [15 29]]	99.34
	Siena	[[2134 0] [16 15]]	99.26

4.2.3 Logistic Regression Model :

After running JMI (Joint Mutual Information) feature selection, top 13 features of both MIT and Siena databases were filtered and later were used for training logistic regression by doing train test split. The logistic regression max_iter was set to 10000 iterations and the model converged under the 70 percent training data of the filtered features. The cross validation was measured across the whole dataset of the filtered features. The accuracy achieved was about 73.9% for the MIT database and 89.2% for the Siena database for top 13 selected features. Random seed was set to 92.

Table: Logistic regression on MIT and Siena data for the top extracted features through various feature selection methods

Classifier	Dataset	Accuracy (%) With Top 3% Features	Accuracy (%) With Top 5% Features	Accuracy (%) With Top 10% Features
Logistic Regression	MIT	73.9%, solver='lbfgs'	72.7%, solver='lbfgs'	62.9%, solver= 'saga'
	Siena	90.9%. solver='lbfgs'	90.9%, solver='sag'	95.5%, solver='sag'

Table: Real Time MIT and Siena Dataset Accuracy with top 5% selected features.

Classifier	Dataset	Confusion Matrix	Accuracy (%) With Top 5%

			Features
MLP Classifier	MIT	[[2196 17] [44 31]]	97.33
	Siena	[[1634 166] [28 49]]	89.66

4.2.4 K-means Clustering model :

After running MIM (Mutual Information Maximization) feature selection, top 25 features of both MIT and Siena databases were filtered and used for implementation of k-means for classification. The model was created by clustering for a binary class system and was scored based on the labels we had on the dataset. According to the documentation, the rand_score looks at the similarity of the two assignments, where 1.0 is a perfect match, and poorly agreeing labels will be closer to 0. Here we were able to achieve a rand_score of around 0.5. Based on the evaluation of the different algorithms on the training database, we determined that K-means was not an effective algorithm for seizure classification. Therefore we are not trying it out on real time dataset.

Classifier	Dataset	Accuracy (%) With Top 25 Features	Accuracy (%) With Top 14 Features	Accuracy (%) With Top 10 Features
K Means	MIT	49.789% k=2	49.39% k=2	49.39 % k=2
	Siena	50.26% k=2	49.64% k=2	49.56% k=2

Classifier	Dataset	Rand_Score With Top 25 Features	Rand_Score With Top 14 Features	Rand_Score With Top 10 Features
K Means	MIT	0.5065 k=2	0.4998 k=2	0.4998 k=2
	Siena	0.4995 k=2	0.4995 k=2	0.5037 k=2

4.2.5- Support Vector Machines

The initial run of SVM models were run using a subset of features determined by the MRMR algorithm. In this case, we selected 10, 20, 25%, and 50% features from each of the datasets. Four different types of SVMs were tested; an SVM using the RBF kernel and grid searched optimal gamma and C values, calculated for each subset of the data tested; a polynomial SVMs of degrees 5 and 6, and a linear SVM. In this, we utilized 5-fold cross validation and a singleton 70% training, 30% testing dataset splits to evaluate the model. A confusion matrix was created from the results of the singleton split so that it could be further analyzed.

The top results from each feature set are shown in the tables below. It is evident that the optimized RBF kernel SVM outperformed other types of SVM, although at lower feature counts showed the benefits of this optimization more so than in higher feature counts. Based on the lightweight nature of the network, along with its extremely close generation gap, we determined that the 20 feature, RBF kernel SVM was the best model found for the MIT dataset, achieving an approximately 90% mean cross validation accuracy, 91.7% training accuracy, and 91.01% testing accuracy. The gamma and C values used were 5.179 and 12.895, respectively. The model was very sensitive, having a 5.6% false negative rate. While this is still high for medical applications, this still shows very good performance.

In the Siena Database, the performance was markedly lower overall, tending to have maximums in the high 70%s and low 80%, with larger generalization gaps, making a worse performance overall. The 10 feature RBF kernel SVM still won out in this case, but only just when compared to other algorithms. The 50% features Linear SVM actually had higher performance and a low generalization gap, but it takes significantly more features, making it less practical. The sensitivity of this model was also much lower, with nearly 15% of the entries being false negatives.

Due to time constraints, we were unable to evaluate the SVM on the real time dataset.

Table: Top scoring SVM classifiers on the MIT and Siena Databases

Classifier	Dataset	Accuracy with Top 10 Features	Accuracy with Top 20 Features	Accuracy with Top 25% of Features	Accuracy with Top 50% of Features
K Means	MIT	88.76%	91.7%	92.3%	96.59%
	Siena	80.17%	76.67% (99%train)	84.26% (99.7% train)	87.75% (99.7% train)

4.2.6- Ensemble model with Maximum Voting (Logistic Regression, MLP, and Decision Tree)

We designed an Ensemble model with 3 model techniques : Logistic Regression, MLP classifier and decision Tree classifier. We took the best 40 features of the MIT dataset to train the ensemble model and 56 best features of the Siena dataset which we found through our feature extraction methods. Using all common and some uncommonly selected features, we tested accuracy on 30 percent of the dataset and achieved quite satisfactory accuracy. We also found that there were more false negatives on the Siena dataset than the MIT dataset that was determined by the ensemble algorithm. The possible reason is we chose more of the selected features for Siena than MIT. As a result, the accuracy dropped. We also tried out the trained ensemble model on the real time MIT dataset, which achieved 98.5% accuracy. Although it got some outputs wrongly as false negative, most of the True positive outputs were correct on the totally unseen dataset.

Table: Ensemble model based on maximum voting for 3 model cases : Logistic Regression, MLP classifier and Decision Tree.

Classifier	Dataset	Accuracy with 70/30 train test split of MIT/Siena dataset	Confusion Matrix For train test split
Ensemble Model with Max voting (Logistic Regression, MLP classifier and Decision Tree)	MIT	99.5 %	[[507 3 2 556]]
	Siena	91.7%	[[135 6 20 154]]
	Real MIT	98.5% (Accuracy on totally new realtime test data)	[[8487 0 129 0]]

5. Timelines

The timeline of this project took place over approximately 2 months. In the first 3 weeks, we arranged all data, ran all feature extraction, and ran all feature selection techniques. In the week following, single-algorithm evaluation took place, and candidates for the ensemble classifier were selected. In week 5, the ensemble classifiers were evaluated and optimized. Week 6 involved testing the final algorithm on simulated real-time data, and the rest of the time was spent on compiling and analyzing results.

6. References

1. Dressler, O et al. “Awareness and the EEG power spectrum: analysis of frequencies”. In: British journal of anaesthesia 93.6 (2004), pp. 806–809.
2. Wen, T., and Zhang, Z. (2017). Effective and extensible feature extraction method using genetic algorithm-based frequency-domain feature search for epileptic EEG
3. Multiclassification. Medicine 96:e6879. doi: 10.1097/MD.0000000000006879
4. Rasekhi, J., Mollaei, M. R., Bandarabadi, M., Teixeira, C. A., and Dourado, A. (2013). Preprocessing effects of 22 linear univariate features on the performance of seizure prediction methods. J. Neurosci. Methods 217, 9–16. doi:10.1016/j.jneumeth.2013.03.019
5. Ren, Y., and Wu, Y. (2014). Convolutional deep belief networks for feature extraction of EEG signal. in International Joint Conference on Neural Networks (Beijing), 2850–2853.

6. Alpaydin E. Introduction to Machine Learning. 2nd ed. MIT Press; Cambridge, MA, USA: 2010. [Google Scholar]
7. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00472-4
8. Livshin, I. (2022). Artificial neural networks with java : tools for building neural network applications (Second). Apress. https://doi.org/10.1007/978-1-4842-7368-5
9. Understanding logistic regression. (2017, May 9). GeeksforGeeks. https://www.geeksforgeeks.org/understanding-logistic-regression/
10. Jones. Creed, personal communication, October 18, 2022
11. Spence, M. (n.d.). What does amplitude mean in EEG? Stamina Comfort. Retrieved December 6, 2022, from https://staminacomfort.com/what-does-amplitude-mean-in-eeg
12. Pyrzowski, J., Le Douget, J.-E., Fouad, A., Siemiński, M., Jędrzejczak, J., & Le Van Quyen, M. (2021). Zero-crossing patterns reveal subtle epileptiform discharges in the scalp EEG. Scientific Reports, 11(1), 4128. https://doi.org/10.1038/s41598-021-83337-3
13. Interspike-intervals—Relation-to-epilepsy-syndrome. (n.d.). AES. Retrieved December 6, 2022, from https://aesnet.org/abstractslisting/interspike-intervals--relation-to-epilepsy-syndrome
14. Brown G, Pocock A, Zhao MJ, Lujan M 2012. Conditional Likelihood Maximization: A Unifying Framework for Information Theoretic Feature Selection. J. Mach. Learn. Res. Vol.13 pp. 27-66
15. Morley A, Hill L, Kaditis AG 2016 10-20 System EEG Placement. European Respiratory Society from https://www.sleep.pitt.edu/wp-content/uploads/2020/03/10-20-system-el.pdf
16. Vijayan, Sujith personal Communication, 12/5/2022