Machine Learning Applications on Neuroimaging for Diagnosis and Prognosis of Epilepsy

Abisha Accamma Vinod
Department of Computer Application
Amal Jyothi College of Engineering
Kanjirappally, India
abishaaccammavinod2022a@mca.ajce.in

T J Jobin
Department of Computer Application
Amal Jyothi College of Engineering
Kanjirappally, India
mailto: tjjobin@amaljyothi.ac.in

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

***Abstract* — Machine learning is becoming more significant in medical image processing, resulting in new improvements in neuroimaging clinical applications. Epileptic seizures are caused by a defect in brain functions, which can have a serious effect on the patient's health. Predicting the commencement of epileptic seizures before they start is very important for medication-assisted seizure prevention. Electroencephalogram (EEG) signals are utilized to predict epileptic seizures using machine learning techniques and computational methodologies.**

I. INTRODUCTION

Epilepsy is a central nervous system (CNS) illness that affects roughly 1.2 percent of the population in the United States (3.4 million individuals) and more than 65 million people worldwide. Furthermore, one out of every 26 people will get epilepsy at some point in their lives. Seizures come in a variety of forms, each with its own set of symptoms, such as loss of consciousness, jerking movements, or bewilderment. Some seizures are significantly more difficult to detect visually; patients frequently show signs like not reacting or staring blankly for a short amount of time. Seizures can strike at any time, causing injuries such as stumbling, biting the tongue, or losing control of one's pee or stool.

As a result, these are some of the reasons why seizure detection is critical for patients who are under medical monitoring and are suspected of having seizures.

This research will employ binary classification algorithms to predict whether a person is having a seizure based on their EEG.

The method for detecting epilepsy that uses:

- Feature Engineering: For our machine learning system, feature engineering is the process of converting categorical or ordinal characteristics into machine-readable numerical variables..
- Binary Classification Algorithms: Machine learning algorithms ranging from Naive Bayes to deep learning networks can tackle binary classification problems. The data volume (number of samples and characteristics) and data quality determine which solution works best in terms of runtime and accuracy (outliers, imbalanced data).

II. LITERATURE REVIEW

Jie Yuan1, Xuming Ran1, Keyin Liu1 proposes two techniques to applying machine learning methods to neuroimaging data: i) traditional machine learning, which combines manual feature engineering and classifiers, and ii) deep learning, which uses convolutional neural networks and autoencoders.

Syed Muhammad Usman,1 Muhammad Usman,1 and Simon Fong2 For epileptic seizures, patients appear to be in a preictal state.To forecast epileptic episodes, machine learning models are utilized. EEG signal capture, signal preprocessing, signal feature extraction, and classification between distinct seizure states are all included in these machine learning models. The goal of the machine learning-based prediction model was to detect predictable states in a timely manner before seizures began.

MohammadKhubeb Siddiqui,[#1] Ruben Morales-Menendez,[1] Xiaodi Huang, developed epilepsy detection model that aim The goal of this study is to give a taxonomy of statistical features and machine learning classifiers—'black-box' and 'non-black-box'—to provide an overview of the great variety of these techniques over the previous few years. The state-of-the-art methods and theories given will provide a thorough grasp of seizure detection and categorization, as well as future research possibilities..

.

III. MOTIVATION

Seizures come in a variety of forms, each with its own set of symptoms, such as loss of consciousness, jerking movements, or bewilderment. Some seizures are significantly more difficult to detect visually; patients frequently show signs like not reacting or staring blankly for a short amount of time. Seizures can strike at any time, causing injuries such as stumbling, biting the tongue, or losing control of one's pee or stool. As a result, these are some of the reasons why seizure detection is critical for patients who are under medical monitoring and are suspected of having seizures.

This research will employ binary classification algorithms to predict whether a person is having a seizure or not. Thus through early detection of epilepsy, we can prevent seizures early on.

IV. METHODOLOGY

The purpose of this paper is to find a Machine Learning model to identify whether the person having epilepsy or not with the dataset. The dataset is divided into three training, validating and testing data sets. The model should be trainedwith more data to get accuracy in learning. The result from this model can be used to create a news classification.

A. **Data Collection:** With 500 patients, the dataset includes 4097 electroencephalogram (EEG) values each subject over 23.5 seconds. The 4097 data points were then separated into 23 chunks per patient, with each chunk corresponding to a single row in the dataset. Each row has 178 readings that are converted into columns; in other words, one second of EEG readings is made up of 178 columns. There are 11,500 rows and 179 columns in all, with the last column providing the patient's status, whether or not the patient is suffering a seizure.

B. **Model Selection:** Model selection refers to the process of determining a single machine learning model for a training dataset from a pool of candidate machine learning models. Model selection is a technique that can be applied to a variety of models (e.g. logistic regression, SVM, KNN, etc.)

1. **KNN –** KNN stands for K Nearest Neighbor, and it's a classification model that divides a sample into classes based on the k samples closest to it. If k=3 and all three samples closest to the sample in question are class 1, the sample in question will also be categorized as class 1. If two of the three samples closest to the one in question are class 1, the sample in question has a 66 percent chance of being

categorized as class 1.

2. **Logistic Regression –** Logistic regression is a extension of ordinary linear models' concepts and abilities. By projecting the sample points onto the line, logistic regression performs a linear function on your input features. The linear function is used form of generalized linear model, which is an by adding the log of likelihood of each sample point and optimizing the log of likelihood, which is the same as maximizing the likelihood, to obtain the best-fitting logistic regression line. In a binary classification model, the best fitting function would anticipate the probability of the positive class to be really near to 1 (100 percent), and the probability of the negative class to be close to 0.
3. **Stochastic Gradient Descent -** Gradient descent is a method for minimizing a large number of loss functions in a variety of models, including linear regression, logistic regression, and clustering models. In a linear regression model, GD plots all potential sums of squared residuals and calculates the slope by taking the derivative of the slope at the chosen point. The lowest slope of the curve is fined by GD, which in linear regression is when the derivative is 0. In other models, gradient descent finds the smallest slope by making big estimates at first and smaller guesses as it gets closer to the minimal value; this makes gradient descent highly useful for solving for where the derivative equals 0.
4. **Naïve Bayes** - The Bayes theorem is used by the naive Bayes classifier to perform classification. It is assumed that if all features are unrelated, the likelihood of seeing them all together is equal to the product of the probabilities of each feature occurring separately. It calculates the likelihood of each feature for each class. If one of our features is weather outlook, and our dependent variable is whether or not we will play golf, we will calculate the likelihood of each weather outlook category based on whether or not we will play.

   We multiply all of these numbers together using these results. This yields a result that represents the probability of X given a class multiplied by a class's probability. P(X|C)P(C). This is done for both classes, and then both sides are divided by P(X) to be normalized. Finally, we assess the normalized probability of scenario X given its class to determine whether or not we will classify a sample under it. The sample will then be categorized under the class with the higher probability.
5. **Decision Tree Classifier** - A decision tree categorizes the sample based on the answer. The classification can be numeric or categorical. The classifying method divides data into sub regions of the same class repeatedly, and the tree ends when the algorithm has divided all samples into pure classes, or when some of the classifier qualities are met. A Root Node, often known as The Root, is the top of the tree. Internal Nodes, or simply Nodes, are nodes that branch out farther; they feature arrows going both to and away from them. Finally, arrows point to Leaf Nodes, or just Leaves, but not away from them.
6. **Random Forest -** By bootstrapping the collection of samples or employing a random amount of features at each split, a random forest is created by bagging decision trees that are not associated with each other. RF combines the simplicity of decision trees with the flexibility of RF, resulting in a significant increase in accuracy, which is one of decision trees' flaws as a classifier.

   Step I: Create a bootstrapped dataset first. We can choose the same sample multiple times.

   Step II: Using the bootstrapped dataset, design a decision tree with only a random subset of variables at each node.

   Step III: Create a new tree and repeat step 1 with a new bootstrapped dataset. Create at least 100 trees if possible.

7. **Gradient Boosting Classifier -** We begin with a leaf that represents an initial forecast for each individual; the log(odds), which is the logistic regression equivalent of the average, is the first prediction for each individual. If four people have diabetes and two do not, the log(odds) is log(4/2) = 0.7, which we place in the first leaf. The log(odds) is then converted into a probability using a logistic function. We can classify everyone in the training dataset as having diabetes because the likelihood of having diabetes is larger than 0.5 (we may also choose another number). However, labelling everyone in the dataset as diabetic is pointless because two persons do not have diabetes.
8. **Extremely Random Trees -** The ExtraTrees Classifier is comparable to the Random Forest Classifier except for the following:

   Instead of bootstrapping samples, samples are obtained from the complete training set when choosing a variable at the split.
   Instead than being specified as in Random Forest, node splits are chosen at random.
9. **XGBoost Classifier** - XGBoost is similar to gradient boosting except

   - Trees have a varying number of terminal nodes
   - Leaf weights of the trees that are calculated with less evidence is shrunk more heavily
   - Newton Boosting provides a direct route to the minima than gradient descent
   - Extra randomisation parameter is used to reduce the correlation between trees

- Uses a more regularized model to control over-fitting since standard GBM has no regularization, which gives it better performance over GBM.
- XGB implements parallel processing, and is much faster than GBM.

C. **Model Evaluation:** Model selection is the process of choosing a single machine learning model for a training dataset from a set of candidate machine learning models. Model selection is a technique that can be applied to a variety of models (e.g. logistic regression, SVM, KNN, etc.)

V. BUILD MODEL

The model building is the main step in the epilepsy detection. While building the model user use the algorithms. The steps involved are:

1. Import the packages that are necessary.

```
import pandas as pd
df = pd.read_csv("epilepsy.csv")
```

```
import warnings
warnings.filterwarnings('ignore')
```

```
from sklearn.decomposition import PCA
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.feature_selection import RFE
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import MinMaxScaler, StandardScaler
```

```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="whitegrid")
```

2. Add the data into a DataFrame,, then get the shape of data.

```
df = pd.read_csv("epilepsy.csv")
df.shape
```

3. Then split the dataset into training and testing datasets

```
df_valid_test = df_data.sample(frac=0.3)
print("Validation/Test Split Size: %.1f" % (len(df_valid_test) / len(df_data)))

df_test = df_valid_test.sample(frac=0.5)
df_valid = df_valid_test.drop(df_test.index)

df_train_all = df_data.drop(df_valid_test.index)
```

4. Then, select the model

```
from sklearn.metrics import roc_auc_score, accuracy_score, precision_score, recall_score
def calc_specificity(y_actual, y_pred, thresh):
    # calculates specificity
    return sum((y_pred < thresh) & (y_actual == 0)) /sum(y_actual ==0)

def print_report(y_actual, y_pred, thresh):

    auc = roc_auc_score(y_actual, y_pred)
    accuracy = accuracy_score(y_actual, (y_pred > thresh))
    recall = recall_score(y_actual, (y_pred > thresh))
    precision = precision_score(y_actual, (y_pred > thresh))
    specificity = calc_specificity(y_actual, y_pred, thresh)
    print('AUC:%.3f'%auc)
    print('accuracy:%.3f'%accuracy)
    print('recall:%.3f'%recall)
    print('precision:%.3f'%precision)
    print('specificity:%.3f'%specificity)
    print('prevalence:%.3f'%calc_prevalence(y_actual))
    print(' ')
    return auc, accuracy, recall, precision, specificity
```

5. Analyze results baseline models

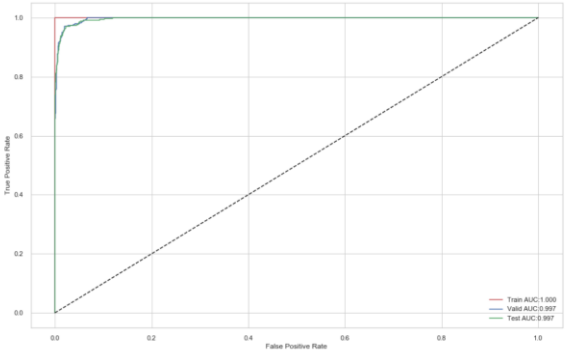
```
df_results = pd.DataFrame({'classifier':['KNN','KNN','LR','LR','SGD','SGD','NB','NB','DT','DT','RF','RF','GB','GB','XGBC','XGBC','ETC','ETC'],
                           'data_set':['train','valid']*9,
                           'auc':[knn_train_auc, knn_valid_auc,lr_train_auc,lr_valid_auc,sgdc_train_auc,sgdc_valid_auc,nb_train_auc,nb_valid_auc,tree_t
                           'accuracy':[knn_train_accuracy, knn_valid_accuracy,lr_train_accuracy,lr_valid_accuracy,sgdc_train_accuracy,sgdc_valid_accura
                           'recall':[knn_train_recall, knn_valid_recall,lr_train_recall,lr_valid_recall,sgdc_train_recall,sgdc_valid_recall,nb_train_re
                           'precision':[knn_train_precision, knn_valid_precision,lr_train_precision,lr_valid_precision,sgdc_train_precision,sgdc_valid_
                           'specificity':[knn_train_specificity, knn_valid_specificity,lr_train_specificity,lr_valid_specificity,sgdc_train_specificity
```

6. Model Evaluation

```
# load the model, columns, and scaler
best_model = pickle.load(open('best_classifier.pkl','rb'))
cols_input = pickle.load(open('cols_input.sav','rb'))
scaler = pickle.load(open('scaler.sav', 'rb'))

# load the data
df_train = pd.read_csv('df_train.csv')
df_valid= pd.read_csv('df_valid.csv')
df_tests pd.read_csv('df_test.csv')

# create X and y matrices
X_train = df_train[cols_input].values
X_valid = df_valid[cols_input].values
X_test = df_test[cols_input].values

y_train = df_train['OUTPUT_LABEL'].values
y_valid = df_valid['OUTPUT_LABEL'].values
y_test = df_test['OUTPUT_LABEL'].values

# transform our data matrices
X_train_tf = scaler.transform(X_train)
X_valid_tf = scaler.transform(X_valid)
X_test_tf = scaler.transform(X_test)
```

VI. RESULT

The result shows the true positive rate and False Positive rate with high accuracy.



VII. CONCLUSION

The best model has a lift performance of 4.3, which means it is 4.3 times better than guessing at random. It also correctly predicts the positive classes in the test set 97.4% of the time. If this model were used to predict whether or not a patient is experiencing a seizure, you may expect it to do well in correctly predicting those who are having one.