# ON-CHIP EPILEPSY DETECTION: WHERE MACHINE LEARNING MEETS PATIENT

***A project report submitted to***

*MALLA REDDY UNIVERSITY*

***in partial fulfillment of the requirements for the award of degree of***

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE & ENGINEERING(AI&ML)**

**Submitted by**

**Bhavith Reddy : 2011CS020443**

***Under the Guidance of***

***N.V.P.R Rajeswari***
***Assistant Professor***

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (AI & ML)**

**MALLA REDDY UNIVERSITY**

(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

2023

MALLA REDDY UNIVERSITY

(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

# COLLEGE CERTIFICATE

This is to certify that this is the bonafide record of the application development entitled, **“On-Chip Epilepsy Detection: Where Machine Learning Meets Patient**” Submitted by **Bhavith Reddy (2011CS020443),** B. Tech III year I semester, Department of CSE (AI&ML) during the year 2022-23. The results embodied in the report have not been submitted to any other university or institute for the award of any degree or diploma.

| | |
|---|---|
| **INTERNAL GUIDE** | **HEAD OF THE DEPARTMENT** |
| **N.V.P.R Rajeswari** | **Dr. Thayyaba Khatoon** |
| **Assistant Professor** | **CSE(AI&ML)** |

# ABSTRACT

Epilepsy is a severe and chronic neurological disorder that affects over 65million people worldwide. Moreover, patient-to patient and age-to-age variation on seizure pattern makes such  detection particularly challenging. This presentation will cover the design strategies of patient-specific epilepsy detection System –on chip(Soc). We will first explore the difficulties, limitations and potential pitfalls in wearable interface circuit design, and strategies to overcome such issues To expand the beneficiary group to even infants, and to effectively adapt to each patient a wearable form – factor, patient-specific system with machine learning is of crucial. Finally , an on-chip epilepsy detection and recording sensor SoC will be presented , which integrates all the components .

The main metrics used are sensitivity, specificity, and accuracy; hence, some papers reviewed were excluded due to insufficient metrics. To evaluate the overall performances of the reviewed papers, a simple mean value of all metrics was used. This review indicates that the system that used a Stockwell transform wavelet variant as a feature extractor and SVM classifiers led to a potentially better result.

# CONTENTS

**CHAPTER NO.** **TITLE** **PAGE NO.**

1. INTRODUCTION
1.1 Problem definition 1
1.2 Objective of project 1
1.3 Limitations of project 1

2. ANALYSIS
2.1 Introduction 2
2.2 Software requirements specifications 2
2.2.1 Software requirements 2
2.2.2 Hardware requirements 2
2.3 Existing system 3
2.4 Proposed system 3
2.5 Modules 3
2.6 Architecture 4

3. DESIGN
3.1 Introduction 5
3.2 DFD/ER/UML diagrams 6
3.3 Data set Description 7
3.4 Data preprocessing techniques 8
3.5 Method & Algorithms 8
3.6 Building a Model 9
3.7 Evaluation 9

4. DEPLOYMENT AND RESULTS
4.1 Introduction 10
4.2 Source Code 11-13
4.3 Final Results 14

5. CONCLUSION
5.1 Project conclusion 15
5.2 Future Scope 15

# Chapter 1

# INTRODUCTION

## 1.1 PROBLEM DEFINITION

Several studies have applied different machine learning algorithms to detect seizures in EEG-data . However, most of the work has been done on patient-specific classifiers . A previous study achieved high accuracy in a patient specific setting were the methods Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) were compared . To our best belief, no other studies in the patient-independent domain have compared these two methods specifically. Therefore, this study is concerned with a comparative analysis between the algorithms SVM and KNN employed for EEG based epileptic seizure identification. Furthermore, this study will focus on patient-independent classifiers as they are more complicated due to the EEG variability .

## 1.2 OBJECTIVE OF PROJECT

The purpose of the study is to train and test Support Vector Machine and K-Nearest Neighbor on EEG correlates containing epileptic seizures and seizure free intervals, in order to detect early onsets of epileptic seizures. Furthermore the study focuses on investigating the generalizability of epileptic seizure models across different patients. This study uses a dataset provided by the Scalp EEG Database .Since only one dataset is used, there could be limitations to the conclusions of this study. As the study wants to investigate the generalizability among different patients, a higher amount of data might be needed to achieve better results. If the study was to be performed on other datasets the results might differ. The scope is also limited by the EEG channels that were chosen. The feature extraction includes selecting a group of channels from each patient, meaning that the patient would not be applicable for this study if EEG data from the specified channels is not available.

## 1.3 LIMITATIONS OF PROJECT

- Time consuming
- Requires large datasets
- Complicated process

# Chapter 2

# ANALYSIS

## 2.1 INTRODUCTION

Epilepsy is characterized by recurrent, unprovoked seizures, which are classified depending on if the onset is partial or generalized. A partial on set means that the epileptic activity begins in one hemisphere of the brain, and a generalized onset is when there is initial involvement of both hemispheres. Depending on where in the brain the epileptic activity begins and how far it spreads, the various seizure types differ. Hence, the seizures alter from brief lapses of attention to severe and long-lasting convulsions . Epilepsy can be assessed by the electroencephalogram (EEG). EEG is one method for measuring electrical activity of the brain, and lately there has been a huge rise of interest in the decoding of brain states based on the underlying EEG.

EEG records the electrical signals sent by the brain through electrodes attached on the head of the subjects and are then sent to a computer for interpretation . Diagnosis of epilepsy based on EEG signals can be troublesome and slow, especially for long-duration EEG signals . It could also be difficult to characterize and interpret the EEG signal, since it is highly non-linear and non-stationary . Nevertheless, it is a well-established technique with low costs associated with it. One important aspect of epilepsy research includes analyzing and classifying EEG-data in order to detect seizures in early stages. If a seizure is detected in its early stages, then neurostimulation can be applied to prevent the seizure from developing and spreading to other parts of the brain. Therefore it is essential to find an efficient method for automatic seizure detection .

## 2.2 SOFTWARE REQUIREMENT SPECIFICATION

### 2.2.1 Software Requirement

- Jupyter Notebook
- Anaconda3
- Google Chrome or Microsoft edge of Latest Version

### 2.2.2 Hardware requirement

- OS: Windows 10 or Higher
- Processor: Intel i5 processor or Higher

- Ram: Minimum 8 GB or Higher
- Hard Drive: Minimum 256 GB or Higher

## 2.3 EXISTING SYSTEM

In the existing system that we are using currently has less number of datasets that make patient to patient and age to age variation on seizure pattern makes the detection challenging**.**

## 2.4 PROPOSED SYSTEMS

In the new proposed system due to numerous datasets present, patient to patient and age to age variation on seizure pattern which was earlier difficult for detection becomes easier. Even if the datasets are limited the D^2 A- SVM(support vector machine) classifier that we are using performs well while minimizing the hardware cost.

## 2.5 MODULES

### Data Pre-processing Module

Data Pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not alwaysa case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data Pre- processing task.

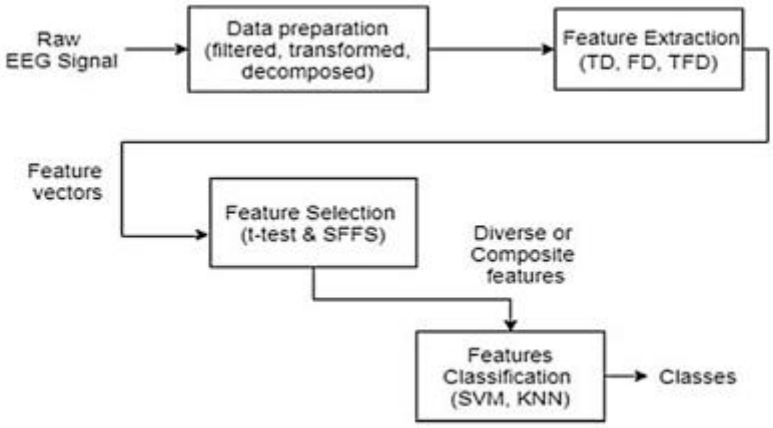
### Data Visualization Module

Visualising the data for data analysis. We can find the relations between the attributes and can work on them to make any necessary changes on our training data.

### Training and Testing

In this Module we will train the system using SVM classifier. Using the training Model the system will produce the classifies the testing data

## 2.6 ARCHITECTURE



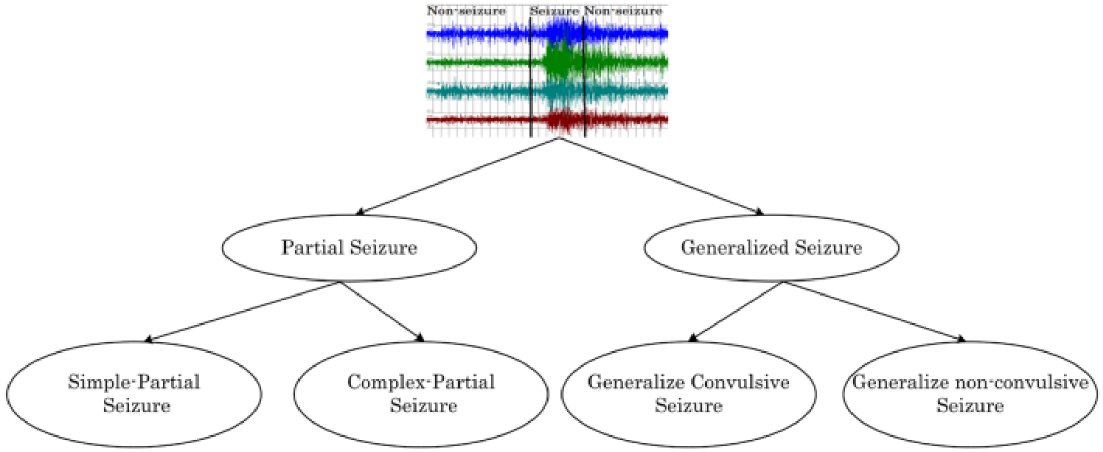


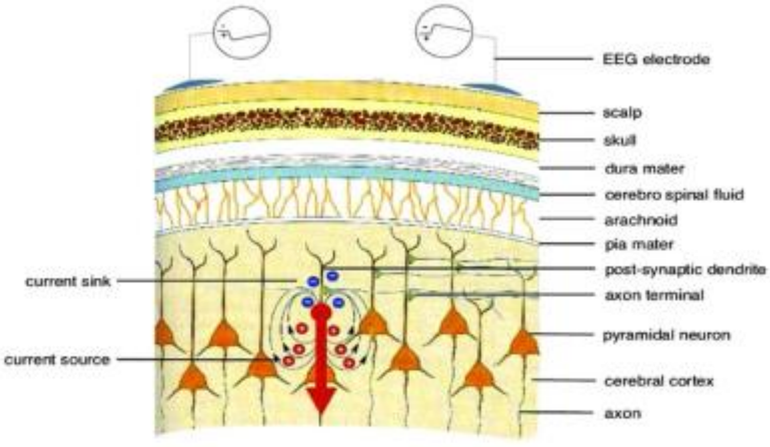
**Fig. 1** Types of seizure. Showing types of seizure and its sub-types

# Chapter 3

# DESIGN

## 3.1 INTRODUCTION

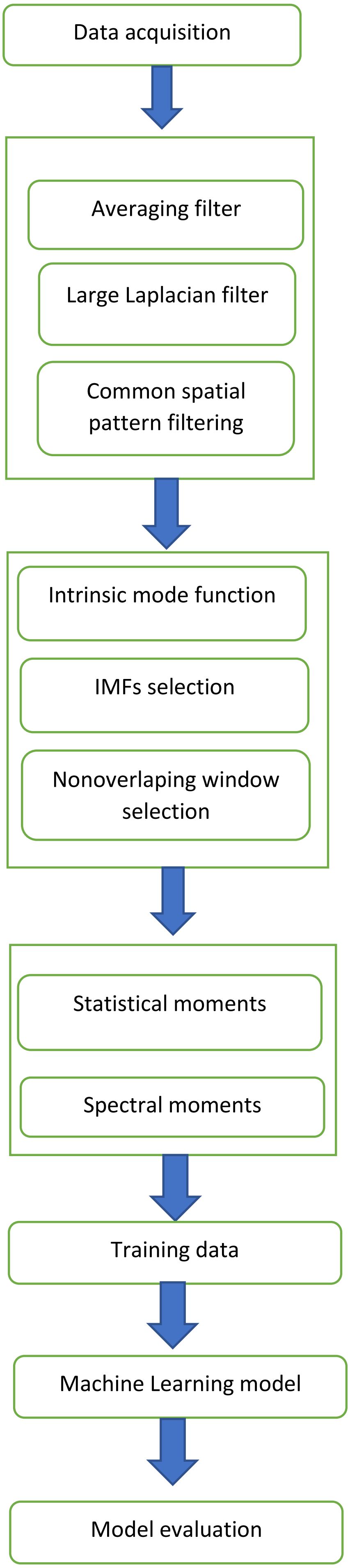
In this study, we explored machine learning approach in the diagnoses and detection of epileptic seizure. Doing this, we used K-nearest neighbors (KNN) and Support Vector Machine (SVM) classification systems. We analyzed the EEG signal in the time and frequency domains and extracted the features of each EEG wave. After comparing the efficiency of each feature individually, we combined the features together and chose the best and most significant features using two different techniques: T-test and Sequential Forward Floating Selection (SFFS). The best selected features were then fed into SVM and KNN classifiers and the results of our models were evaluated using k- fold cross-validation methodology.



(figure showing the Diagram of neurons firing during EEG)

Detecting the appearance of preictal state predicts the seizure. Therefore, the purpose of our investigation is to detect the appearance of preictal state for epileptic seizures. Machine learning models are used to predict epileptic seizures. These machine learning models include EEG signal acquisition, signal preprocessing, features extraction from the signals, and finally classification between different seizure states. The objective of the prediction model with machine learning was to detect preictal state's sufficient time before seizure onset starts . However, enough time for the predictive preictal state of the gland and maximum sensitivity are important, and they remain as a performance issue in the prediction of epileptic seizures

### 3.2 UML diagram



## 3.2 Data set description

This data contains EEG signal of five healthy participants as well as five patients who were diagnosed with epilepsy. Two resting situations of eyes open and eyes closed were used to record the brain EEG signal for healthy subjects. The standard 10–20 scheme was used to place the electrodes on the subjects' scalp and the signal recorded continuously. The dataset consists of 5 folders (A–E) and each folder is made of 100 single channel EEG segments which was recorded at the sampling rate of 173.61 Hz and band-pass filter setting of 0.5340 Hz.

For data transformation, data processing is a decisive step to extract meaningful information from the collected raw dataset. As such, diferent feature extraction techniques have been used; as shown in Table 1. Tese methods are generally applied to the extracted EEG signal dataset . The raw dataset becomes rich in terms of diferent statistical measure values

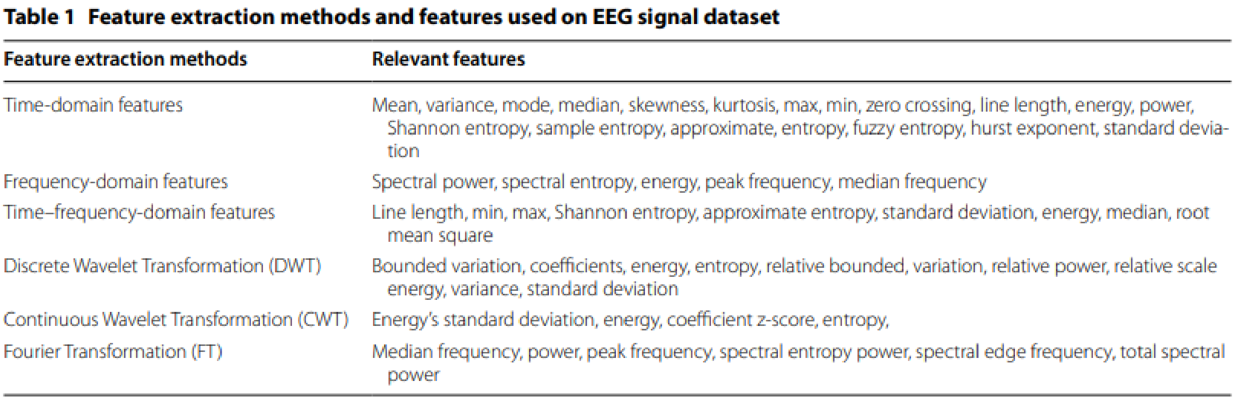


**Table 1 Feature extraction methods and features used on EEG signal dataset**

| Feature extraction methods | Relevant features |
|---|---|
| Time-domain features | Mean, variance, mode, median, skewness, kurtosis, max, min, zero crossing, line length, energy, power, Shannon entropy, sample entropy, approximate, entropy, fuzzy entropy, hurst exponent, standard deviation |
| Frequency-domain features | Spectral power, spectral entropy, energy, peak frequency, median frequency |
| Time–frequency-domain features | Line length, min, max, Shannon entropy, approximate entropy, standard deviation, energy, median, root mean square |
| Discrete Wavelet Transformation (DWT) | Bounded variation, coefficients, energy, entropy, relative bounded, variation, relative power, relative scale energy, variance, standard deviation |
| Continuous Wavelet Transformation (CWT) | Energy's standard deviation, energy, coefficient z-score, entropy, |
| Fourier Transformation (FT) | Median frequency, power, peak frequency, spectral entropy power, spectral edge frequency, total spectral power |

The data set were chosen since it is one of the biggest that is publicly available. It consists of more than 20 patients, while other sets only had around five patients and not as much EEG-data. A study where patient-specific classifiers were investigated used this data set, claiming that it is harder to achieve high accuracy with cross-patient classifiers. This was also a reason for the data set to be chosen as the result from this cross-patient study then could be compared to the patient-specific study

## 3.4 Data Pre-Processing Techniques

- Getting the dataset
- Importing libraries
- Importing datasets
- Finding missing data
- Encoding dataset into test and training set
- Feature scaling

## 3.5 Methods & Algorithm

The next step after the feature extraction is to apply classification techniques to the extracted features. The techniques utilized in this study was the Support Vector Machine and K-Nearest Neighbor. Both of the methods were chosen for this study since there are a lot of related work that have used them. In section 2.5, three studies that used SVM were described, which both achieved high accuracies. One of these studies focused on patient non-specific classification, which this study also wants to investigate. It is convenient to have other studies that utilized the same techniques as in this one, to be able to compare the results achieved with each other

The Support Vector Machine for Binary Classification provided by MATLAB was used in the study. As the data is highly non-stationary and not linearly separable a rbf kernel was used for the SVM.

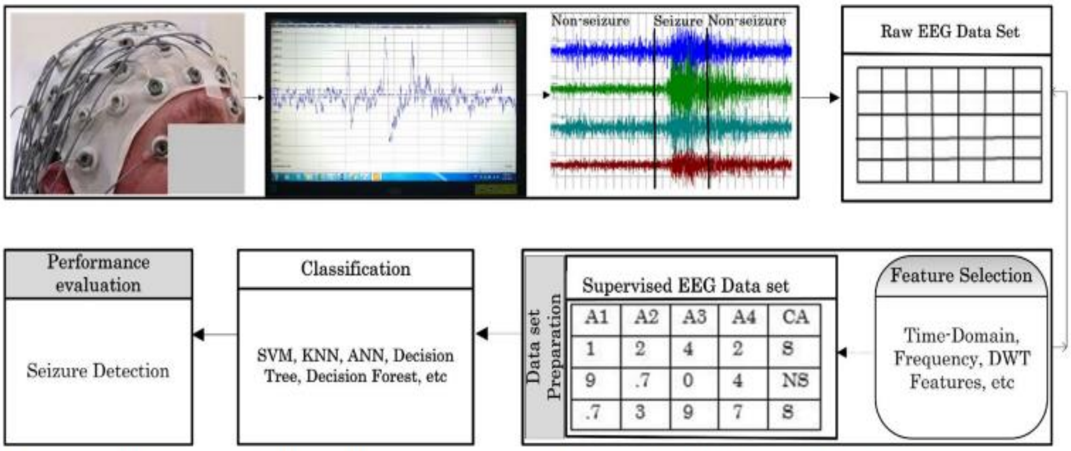
The value of K was set to the square root of the total number of data points in the data set. The values were K = 181 for set A, K = 91 for set B and K = 49 for set C. For all of the sets the distance function was set to euclidean.

To validate the results of the two methods, cross validation was used. More specifically the data extracted from each patient was tested on classifiers trained on data from all other patients, meaning that we have a 21 fold cross validation where the number of folds is equivalent with the number of patients. The results from each fold in terms of accuracy and latency is then averaged in single values for both SVM and KNN.

As data was collected from the same subjects for both of the machine learning models, a paired sample t-test was used for each of the data sets to determine if the accuracies produced by the SVM and KNN classifiers were statistically significantly different. The null hypothesis was that the pairwise differences between the results from SVM and KNN were equal to zero. The chosen significance level was 0.05. Similarly four paired sample t-tests were made for each of the time stamps measured for latency.

## 3.6 Building a model

A feature extraction was performed to transform the EEG signals into values that later was used in the classification. The extracted features intend to minimize the loss of significant information embedded in the signals. For this study the frequency domain features were extracted. A script was made in MATLAB to perform the feature extraction. The files containing the digitized EEG signals were read with an reader. Since the files contained EEG data from at least one hour, the recordings were divided into specified epochs to make the data more manageable and to separate the data according to the type of activity (seizure or non-seizure).



| A1 | A2 | A3 | A4 | CA |
|---|---|---|---|---|
| 1 | 2 | 4 | 2 | S |
| 9 | .7 | 0 | 4 | NS |
| .7 | 3 | 9 | 7 | S |

**Fig. 2** Basic model of epileptic seizure detection. This explains the basic steps to collect the dataset by EEG medium, display of raw EEG signals, transform EEG signals to two-dimensional table, feature selection, prepare the dataset with *seizure (S)* and *non-seizure (NS)*, apply machine learning classifier(s) and seizure detection, or other related tasks

## 3.6 Evaluation

The clinical employment of ES prediction methods requires a sufficient performance and quality check and different evaluation metrics have been proposed in the ES prediction literature. For instance, Osorio et al. proposed sensitivity and false prediction rate as performance parameters of ES predictors [87]. Sensitivity is measured as the ratio of correctly predicted seizures to all seizures. Moreover, contrary to the ideal situation, one can not prevent false prediction and with the increase in sensitivity, the false prediction rate also increases.

The widely used evaluation metrics are described below.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

# Chapter 4

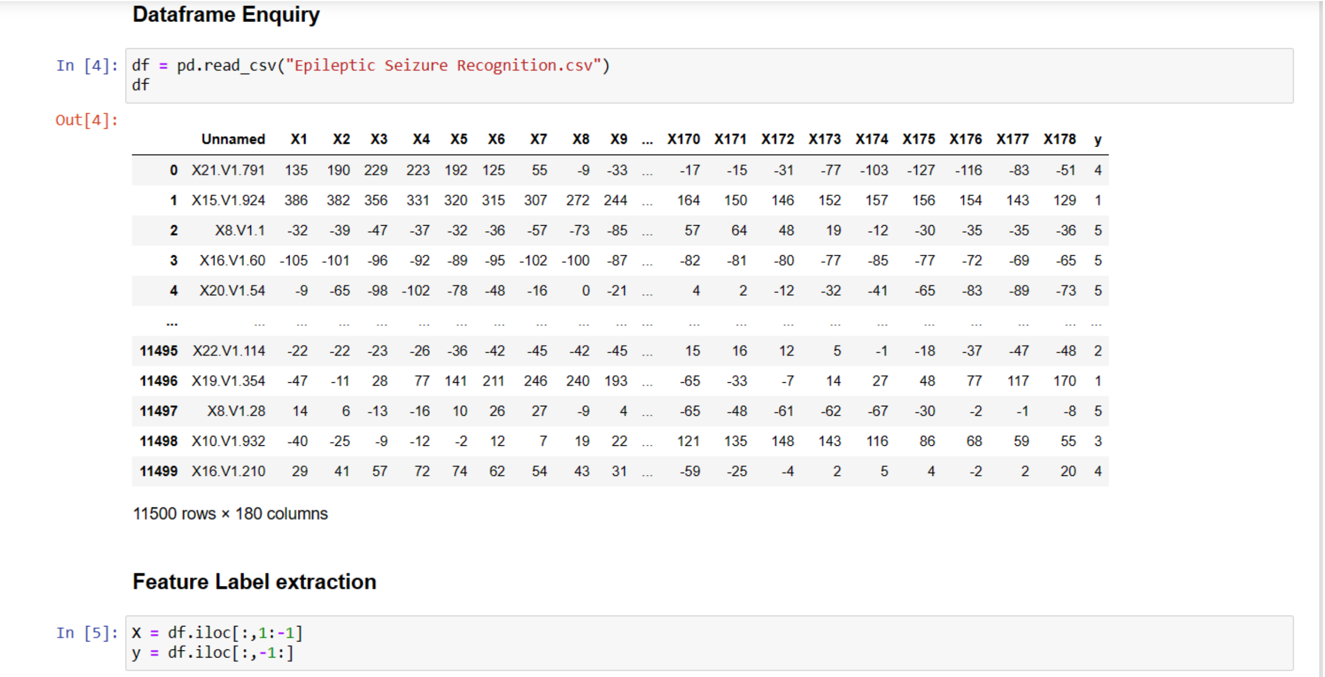
# DEPLOYMENT AND RESULTS

## 4.1 Introduction

Electroencephalography (EEG) is a particularly effective diagnostic tool to study the functional anatomy of the brain during an ES attack. The prediction and medication of epilepsy have been broadly studied through EEG. EEG signals, which are non-Gaussian and non-stationary, measure the electrical activity in the brain which are in turn used to diagnose the type of the brain disorders.

Although there exist several reviews that specifically cover epilepsy seizure prediction using EEG signals, to the best of our knowledge, there does not yet exist a review that covers in depth the application of ML methods for predicting epileptic seizures. For instance, having provided an overview of the evolution of seizure predicting methods since the 1970s till 2006  and have covered the major issues related to methodology of ES prediction. Neuroscience is the multidisciplinary study of the brain. It integrates multifarious disciplines including neuroanatomy (in which neuroanatomists engage with the structures of the human brain), neurochemistry (where chemists observe the chemical properties of intercommunication in the brain), neurophysiology (where the neurophysiologists investigate the electrical properties of the brain) and neuropsychology (where psychologists endeavor to interpret the cognitive domains and the structures that sustain those cognitive domains in neuroscience). Neuroscience also has further divisions for e.g, molecular neuroscience, cognitive neuroscience , clinical neuroscience, computational neuroscience, developmental neuroscience, and cultural neuroscience, to name just a few.

## 4.2 Source Code



### Dataframe Enquiry

```
In [4]: df = pd.read_csv("Epileptic Seizure Recognition.csv")
        df
```

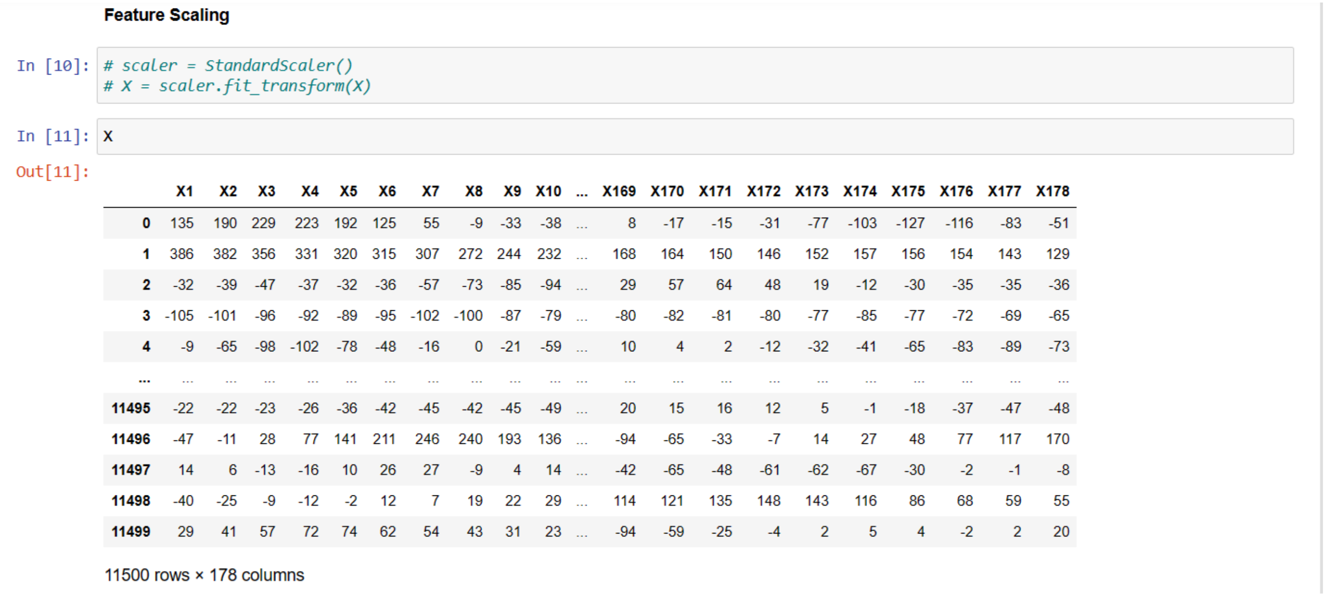
Out[4]:

| | Unnamed | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | ... | X170 | X171 | X172 | X173 | X174 | X175 | X176 | X177 | X178 | y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | X21.V1.791 | 135 | 190 | 229 | 223 | 192 | 125 | 55 | -9 | -33 | ... | -17 | -15 | -31 | -77 | -103 | -127 | -116 | -83 | -51 | 4 |
| **1** | X15.V1.924 | 386 | 382 | 356 | 331 | 320 | 315 | 307 | 272 | 244 | ... | 164 | 150 | 146 | 152 | 157 | 156 | 154 | 143 | 129 | 1 |
| **2** | X8.V1.1 | -32 | -39 | -47 | -37 | -32 | -36 | -57 | -73 | -85 | ... | 57 | 64 | 48 | 19 | -12 | -30 | -35 | -35 | -36 | 5 |
| **3** | X16.V1.60 | -105 | -101 | -96 | -92 | -89 | -95 | -102 | -100 | -87 | ... | -82 | -81 | -80 | -77 | -85 | -77 | -72 | -69 | -65 | 5 |
| **4** | X20.V1.54 | -9 | -65 | -98 | -102 | -78 | -48 | -16 | 0 | -21 | ... | 4 | 2 | -12 | -32 | -41 | -65 | -83 | -89 | -73 | 5 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **11495** | X22.V1.114 | -22 | -22 | -23 | -26 | -36 | -42 | -45 | -42 | -45 | ... | 15 | 16 | 12 | 5 | -1 | -18 | -37 | -47 | -48 | 2 |
| **11496** | X19.V1.354 | -47 | -11 | 28 | 77 | 141 | 211 | 246 | 240 | 193 | ... | -65 | -33 | -7 | 14 | 27 | 48 | 77 | 117 | 170 | 1 |
| **11497** | X8.V1.28 | 14 | 6 | -13 | -16 | 10 | 26 | 27 | -9 | 4 | ... | -65 | -48 | -61 | -62 | -67 | -30 | -2 | -1 | -8 | 5 |
| **11498** | X10.V1.932 | -40 | -25 | -9 | -12 | -2 | 12 | 7 | 19 | 22 | ... | 121 | 135 | 148 | 143 | 116 | 86 | 68 | 59 | 55 | 3 |
| **11499** | X16.V1.210 | 29 | 41 | 57 | 72 | 74 | 62 | 54 | 43 | 31 | ... | -59 | -25 | -4 | 2 | 5 | 4 | -2 | 2 | 20 | 4 |

11500 rows × 180 columns

### Feature Label extraction

```
In [5]: X = df.iloc[:,1:-1]
        y = df.iloc[:,-1:]
```



#### Feature Scaling

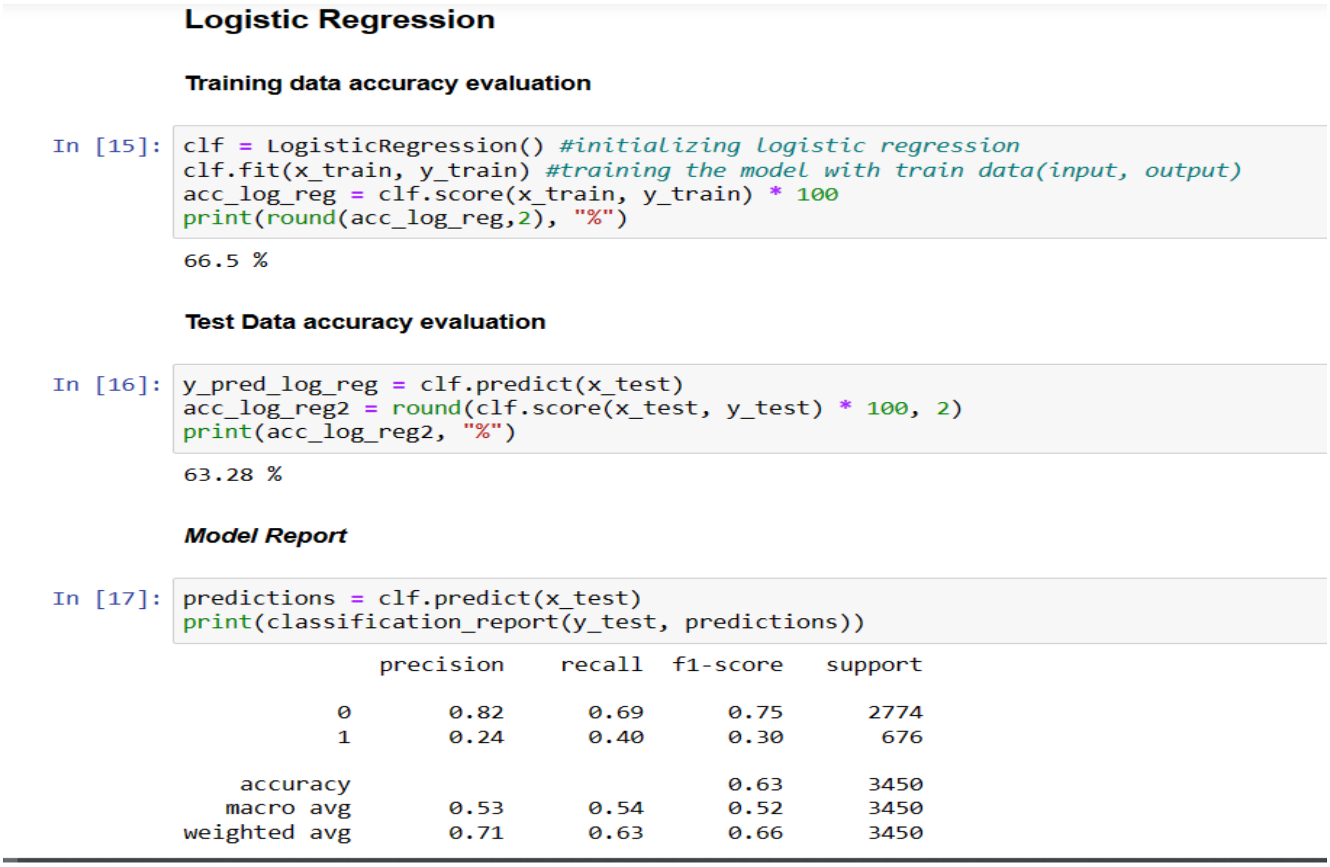
```
In [10]: # scaler = StandardScaler()
         # X = scaler.fit_transform(X)

In [11]: X
```

Out[11]:

| | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | ... | X169 | X170 | X171 | X172 | X173 | X174 | X175 | X176 | X177 | X178 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 135 | 190 | 229 | 223 | 192 | 125 | 55 | -9 | -33 | -38 | ... | 8 | -17 | -15 | -31 | -77 | -103 | -127 | -116 | -83 | -51 |
| **1** | 386 | 382 | 356 | 331 | 320 | 315 | 307 | 272 | 244 | 232 | ... | 168 | 164 | 150 | 146 | 152 | 157 | 156 | 154 | 143 | 129 |
| **2** | -32 | -39 | -47 | -37 | -32 | -36 | -57 | -73 | -85 | -94 | ... | 29 | 57 | 64 | 48 | 19 | -12 | -30 | -35 | -35 | -36 |
| **3** | -105 | -101 | -96 | -92 | -89 | -95 | -102 | -100 | -87 | -79 | ... | -80 | -82 | -81 | -80 | -77 | -85 | -77 | -72 | -69 | -65 |
| **4** | -9 | -65 | -98 | -102 | -78 | -48 | -16 | 0 | -21 | -59 | ... | 10 | 4 | 2 | -12 | -32 | -41 | -65 | -83 | -89 | -73 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **11495** | -22 | -22 | -23 | -26 | -36 | -42 | -45 | -42 | -45 | -49 | ... | 20 | 15 | 16 | 12 | 5 | -1 | -18 | -37 | -47 | -48 |
| **11496** | -47 | -11 | 28 | 77 | 141 | 211 | 246 | 240 | 193 | 136 | ... | -94 | -65 | -33 | -7 | 14 | 27 | 48 | 77 | 117 | 170 |
| **11497** | 14 | 6 | -13 | -16 | 10 | 26 | 27 | -9 | 4 | 14 | ... | -42 | -65 | -48 | -61 | -62 | -67 | -30 | -2 | -1 | -8 |
| **11498** | -40 | -25 | -9 | -12 | -2 | 12 | 7 | 19 | 22 | 29 | ... | 114 | 121 | 135 | 148 | 143 | 116 | 86 | 68 | 59 | 55 |
| **11499** | 29 | 41 | 57 | 72 | 74 | 62 | 54 | 43 | 31 | 23 | ... | -94 | -59 | -25 | -4 | 2 | 5 | 4 | -2 | 2 | 20 |

11500 rows × 178 columns



## Logistic Regression

### Training data accuracy evaluation

```
In [15]: clf = LogisticRegression() #initializing logistic regression
         clf.fit(x_train, y_train) #training the model with train data(input, output)
         acc_log_reg = clf.score(x_train, y_train) * 100
         print(round(acc_log_reg,2), "%")
```

```
66.5 %
```

### Test Data accuracy evaluation

```
In [16]: y_pred_log_reg = clf.predict(x_test)
         acc_log_reg2 = round(clf.score(x_test, y_test) * 100, 2)
         print(acc_log_reg2, "%")
```

```
63.28 %
```

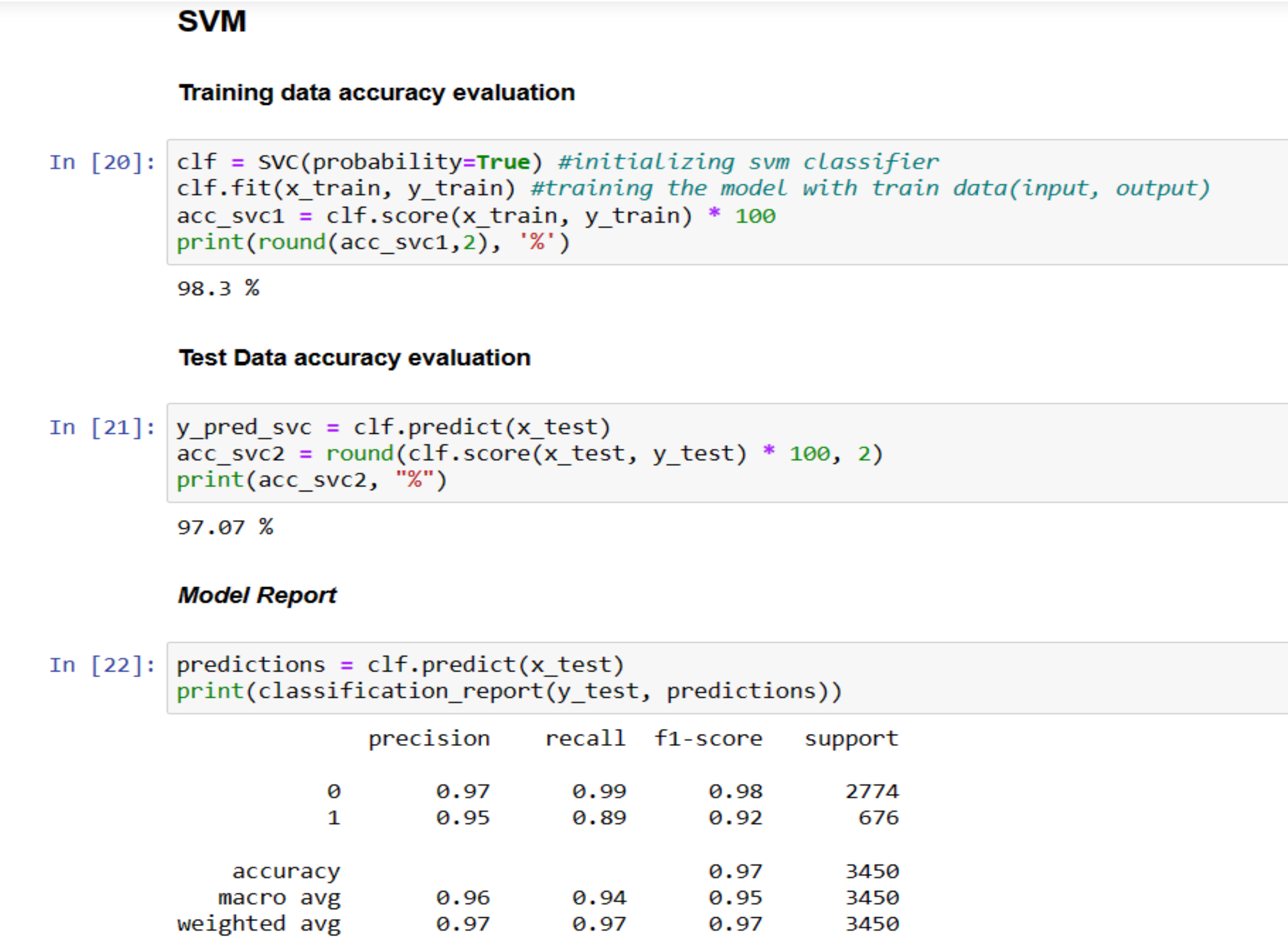
### *Model Report*

```
In [17]: predictions = clf.predict(x_test)
         print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       0.82      0.69      0.75      2774
           1       0.24      0.40      0.30       676

    accuracy                           0.63      3450
   macro avg       0.53      0.54      0.52      3450
weighted avg       0.71      0.63      0.66      3450
```



## SVM

### Training data accuracy evaluation

```
In [20]: clf = SVC(probability=True) #initializing svm classifier
         clf.fit(x_train, y_train) #training the model with train data(input, output)
         acc_svc1 = clf.score(x_train, y_train) * 100
         print(round(acc_svc1,2), '%')
```

```
98.3 %
```

### Test Data accuracy evaluation

```
In [21]: y_pred_svc = clf.predict(x_test)
         acc_svc2 = round(clf.score(x_test, y_test) * 100, 2)
         print(acc_svc2, "%")
```

```
97.07 %
```

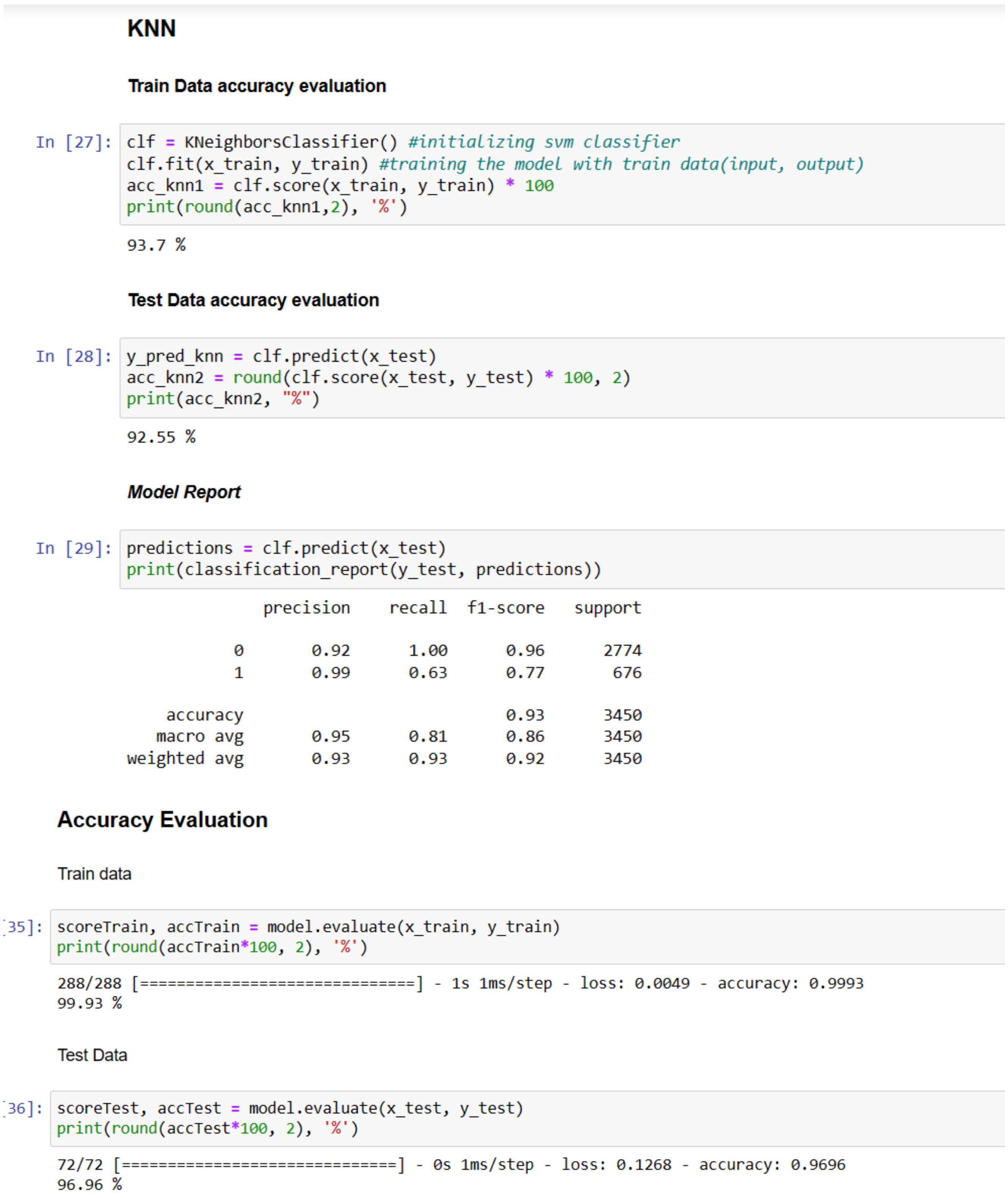
### *Model Report*

```
In [22]: predictions = clf.predict(x_test)
         print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       0.97      0.99      0.98      2774
           1       0.95      0.89      0.92       676

    accuracy                           0.97      3450
   macro avg       0.96      0.94      0.95      3450
weighted avg       0.97      0.97      0.97      3450
```



## KNN

### Train Data accuracy evaluation

```
In [27]: clf = KNeighborsClassifier() #initializing svm classifier
         clf.fit(x_train, y_train) #training the model with train data(input, output)
         acc_knn1 = clf.score(x_train, y_train) * 100
         print(round(acc_knn1,2), '%')
```

```
93.7 %
```

### Test Data accuracy evaluation

```
In [28]: y_pred_knn = clf.predict(x_test)
         acc_knn2 = round(clf.score(x_test, y_test) * 100, 2)
         print(acc_knn2, "%")
```

```
92.55 %
```

### *Model Report*

```
In [29]: predictions = clf.predict(x_test)
         print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       0.92      1.00      0.96      2774
           1       0.99      0.63      0.77       676

    accuracy                           0.93      3450
   macro avg       0.95      0.81      0.86      3450
weighted avg       0.93      0.93      0.92      3450
```

## Accuracy Evaluation

Train data

```
35]: scoreTrain, accTrain = model.evaluate(x_train, y_train)
     print(round(accTrain*100, 2), '%')
```

```
288/288 [==============================] - 1s 1ms/step - loss: 0.0049 - accuracy: 0.9993
99.93 %
```

Test Data

```
36]: scoreTest, accTest = model.evaluate(x_test, y_test)
     print(round(accTest*100, 2), '%')
```

```
72/72 [==============================] - 0s 1ms/step - loss: 0.1268 - accuracy: 0.9696
96.96 %
```

## 4.3 Final Results

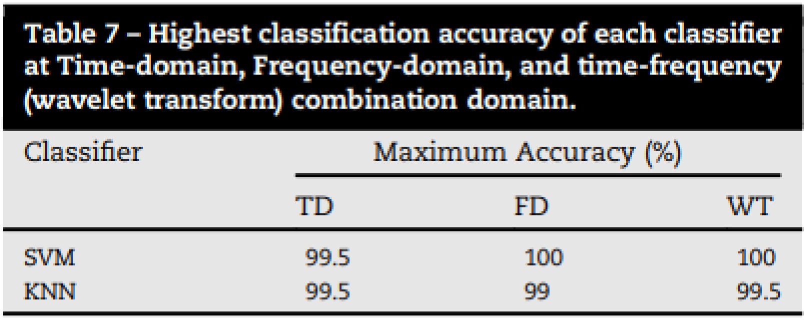
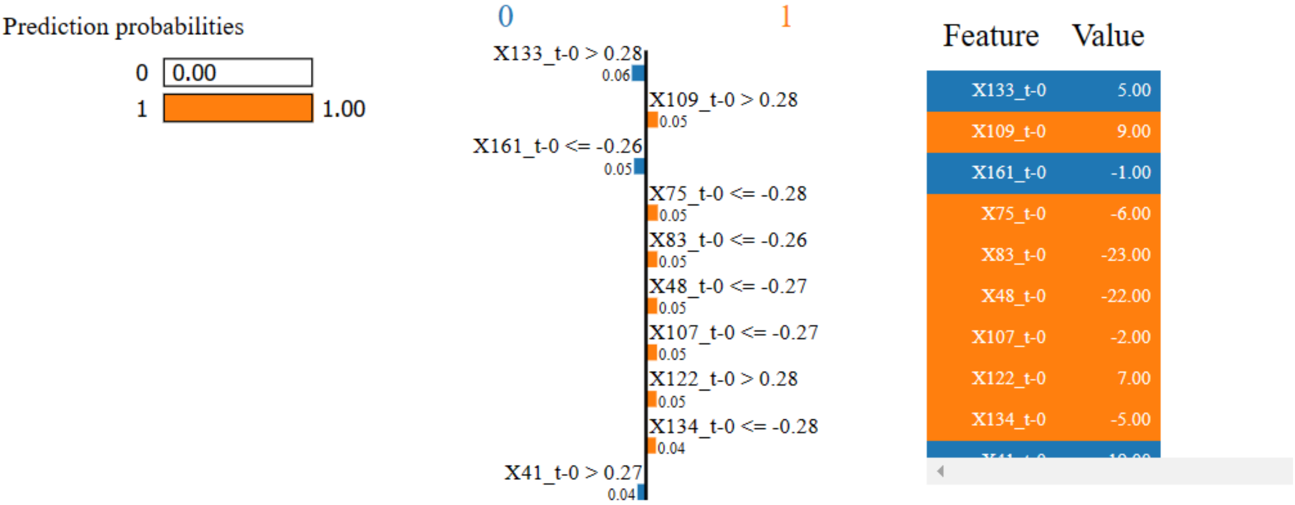


Table 7 – Highest classification accuracy of each classifier at Time-domain, Frequency-domain, and time-frequency (wavelet transform) combination domain.

| Classifier | Maximum Accuracy (%) | | |
|---|---|---|---|
| | TD | FD | WT |
| SVM | 99.5 | 100 | 100 |
| KNN | 99.5 | 99 | 99.5 |



| Feature | Value |
|---|---|
| X133_t-0 | 5.00 |
| X109_t-0 | 9.00 |
| X161_t-0 | -1.00 |
| X75_t-0 | -6.00 |
| X83_t-0 | -23.00 |
| X48_t-0 | -22.00 |
| X107_t-0 | -2.00 |
| X122_t-0 | 7.00 |
| X134_t-0 | -5.00 |
| X41_t-0 | 10.00 |

The results presented in table 4.4 are averaged results and does not show details of how the results of each patient are distributed. To illustrate this a box plot was made  which shows the distribution of the accuracy achieved by SVM and KNN for all training sets and patients. As seen in the plot and as expected the accuracy varies considerably between patients, where the outliers probably represent patients with seizures-patterns that does not have a lot in common with the other patients. If the training data sets had been bigger, the outliers could have been removed to improve the performance of the classifiers. However in this study removing outliers leads to removing a substantial amount of data

# Chapter 5

# CONCLUSION

The two methods, SVM and KNN performed very similar in terms of accuracy and latency. KNN was statistically significantly better than SVM with an accuracy of 78.5% on data that had been scaled for more even proportions between non-seizure and seizure samples. In terms of latency, no statistically significant difference was detected between the two methods. To conclude, KNN performed better in detecting epileptic seizures from EEG data when tested on scaled data, although in practice a larger set of patients would be needed to further differentiate between the two methods.

With the increase of epilepsy, its accurate detection becomes increasingly important. A major challenge is to detect seizures correctly from a large volume of data. Due to the complexity of EEG signals in such datasets, machine learning classifers are suitable for accurate seizure detection. Selecting suitable classifers and features are, however, crucial.

## 5.1 FUTURE SCOPE

To further enhance the research, the amount of data and number of patients could be increased. The amount of data used in this study is quite limited from a machine learning perspective and therefore the classifiers would probably benefit from more data to train on. Due to the variability and differences regarding epileptic seizures, having data from more patients could lead to a more complete detection method.

Furthermore, it is observed in the study that the methods of selecting channels to include have been investigated intensively, thus another important aspect is to consider different channel selection techniques. A more motivated choice of channels could help optimize the classification and hopefully improve the results.