|  |  |
| --- | --- |
| **作品名称** | 癫痫病预测系统 |
| **作品关键词** | 癫痫病 、STM32、机器学习、特征提取、分类、时间序列 |
| **作品简介** | 作品是基于机器学习的癫痫病预测系统， 该系统首先通过Fourier、PCA降维和提取各通道之间的相关性系数对EEG特征提取，然后利用随机森林、SVM、逻辑回归算法和决策树算法对发作前期和发作间期进行分类，从而获得基于人工智能机器学习预测癫痫病发作概率的模型。在测试时，通过电极获取病人脑电波，经蓝牙从机传输信号给蓝牙主机，蓝牙主机获得的脑电波信号通过STM32单片机串口传输到电脑上位机，上位机是采用Labview设计开发的可视化界面，最后在上位机中可显示癫痫病发作概率和脑电波波形。 |

目录

[一、项目概述 5](#_Toc485330522)

[1. 研究背景 5](#_Toc485330523)

[2. 研究目的 5](#_Toc485330524)

[二、可行性分析 6](#_Toc485330525)

[1. 设计方案 6](#_Toc485330526)

[1.1获取数据 6](#_Toc485330527)

[1.2特征提取 6](#_Toc485330528)

[1.3分类 7](#_Toc485330529)

[1.4 脑电波获取仪器（Neuro Sky TGAM） 9](#_Toc485330530)

[1.5 软件开发 10](#_Toc485330531)

[1.6模型评估图 10](#_Toc485330532)

[2. 软件代码 10](#_Toc485330533)

[3. 实物照片 16](#_Toc485330534)

[4. 创新点 16](#_Toc485330535)

[5. 实用性 17](#_Toc485330536)

[6. 市场应用前景 17](#_Toc485330537)

# 一、项目概述

## 1. 研究背景

据世界卫生组织调查表明，目前全球癫痫患者约有7000万，而中国就有1000万左右，且每年还会新增45万左右癫痫患者。由此，抗癫痫活动目前已经成为一项全球性事业。

癫痫病是以脑部神经元过度放电所致的突然、反复和短暂的中枢神经系统功能失常为特征的慢性脑部疾患，俗称“羊癫疯”、“羊羔疯”，病因几乎涉及到中枢神经系统的各类病变，对癫痫的诊断主要是依据对发作症状的详细观察和描述，因此其诊断需要依赖脑电波。但是一方面由于人的大脑过于复杂，人类对它的研究还比较初级，另一方面缺乏足够长的时间记录以及足够多的数据以便研究人员可以运用算法，使癫痫发作预测一直受到阻碍，大多数科研机构仍然停留于实验室阶段。

癫痫的病因及发病机理复杂，对许多患者来说，可以服用高剂量抗惊厥药物以防止癫痫发作，但患者常常会出现副作用。而对于部分癫痫患者来说，药物是无效的，甚至在进行外科手术后，许多患者仍然会有自发性的癫痫发作。这些都将使癫痫患者由于有可能出现的症状而持续焦虑，对个人、家庭、社会造成极大的危害和影响，所以如果可以能够智能、准确的检测癫痫放电对癫痫患者来说具有划时代的意义，同样也减少了对经验必须足够丰富的医生的过度依赖。癫痫病人的发作间期和发作期对于确定癫痫病具有极其重要的价值。

## 2. 研究目的

癫痫作为一种常见的神经系统疾病，具有突发性、暂时性、反复性等特点，对个人、家庭及社会均会造成极大的危害和影响。目前，药物治疗仍然是其主要治疗方法，但采用药物治疗一方面为了要防止癫痫发作需要长期服用高剂量抗惊厥药物而这势必会有副作用，对癫痫患者造成难以忍受的不良反应。另一方面不能及时、准确的诊断及合理规范的治疗，反而更加危害癫痫患者的身心健康甚至于因为癫痫的突然发作造成病人的死亡，同时也减少了癫痫患者对可能出现的症状而持续惧怕、焦虑甚至于对生活丧失信心。而如果可以提前预测癫痫病，病人就可以避免潜在的危险活动，而在需要的时候才使用药物来预防即将发作的癫痫，可以减少副作用。

# 二、可行性分析

## 1. 设计方案

### 1.1获取数据

人的脑电波按频率大致可以分为等四类类型的波。在不同的精神状况下不同的波形占据主导，根据这一原理，我们首先获取大量脑电波数据对数据进行相关性分析。由于缺乏足够长的时间记录以及足够多的数据以便研究人员可以运用算法，癫痫发作预测一直受到阻碍，经过长期、大量的地毯式搜索，我们从相关网站上获取了大量的癫痫患者发病前期、发病间期的脑电图数据。接着我们准备从时间序列数据分析方面入手对获取的癫痫患者脑电波数据进行特征提取与分类。

影响癫痫患者EEG的阳性率的因素有很多，其中人为因素占较大比重。包括脑电波记录时间过短（常规脑电波最少记录时间最少203Min）、记录电极数目过少，导致通过脑电波数据对癫痫病人的预测精度不准。而且目前现有的脑电波数据只有大型医院才有，一般的小科研不容易获取大量脑电波数据，我们经过了较长时间搜索在相关网站上获取了部分病人的脑电波数据。人体的大脑活动的时间状态可以分为Interictal、Preictal、ictal、Post-icta4个状态，我们主要是通过机器学习首先区分Interictal和Preictal来识别出癫痫发作的大脑状态，接着用我们的脑电波获取仪器获取癫痫患者脑电波数据，从而进行分析、对比，使患者可以避免潜在的危险的活动，而在需要的时候才使用药物来预防即将发作的癫痫，从而减少副作用。

### 1.2特征提取

特征提取是指在数据分类过程中，为了实现模式内距离最小化而对原始的数据信息进行的提炼。之所以要进行特征提取，是因为在我们获取的大量脑电波数据中，一个癫痫患者的一个发病前期和发病间期脑电波数据就有300多万个，数据量非常庞大、维数很高，为了更好的区分发病前期和发病间期，需要对数据进行降维，这样通过特征提取，把它们的某些特征提取出来，把那些反应关键数字的特征输入到神经网络中，这样更加有利于网络训练和测试的时间。

#### 1.2.1特征提取方法

（1）采用快速傅里叶变换对脑电波数据进行预处理，在不损失有效信息的前提下简化样本数据复杂程度；

时间序列 FFT 对频率进行切片处理 取对数

快速Fourier变换应用于每1s的数据片段，在频率范围为1-47HZ取对数，相位信息就会被丢弃，在这个阶段，每个训练实例的输出都是形状（N,15），其中N是一个给定病人使用的脑电图通道的数目。在尝试维度减少的同时，选择了1-47HZ的范围，结果比使用其他范围得到了更好的结果。

在时间和频率上计算相关系数及其特征值并附加到FFT数据以形成特征集，然后对特征集使用具有3000棵树的随机森林分类算法进行训练。通过重实验，确定了模型中使用的特性。在时域内的相关系数及其对应的特征值是癫痫发作的特征，可以在时域与频域中运用。

相关系数和频域特征值：

1-47HZ的FFT 标准化 相关系数 特征值

以形状（N，47）作为FFT的输出，然后通过频率进行标准化，这个阶段的输出是相关系数矩阵的上直角三角形，而特征值是按大小排列的。

相关系数和时域特征值：

时间系列 标准化 相关系数 特征值

以形状（N，47）作为FFT的输出，然后通过频率进行标准化，数据的输出是原始数据的时间序列源数据，相关系数的上三角和已排列的特征值都被用作特征。

（2）利用PCA降维对数据进行特征提取。

（3）时间序列数据是一类重要的具有时序特征的数据对象，一个时间序列是一组按照时间先后顺序排列好的数据采样观测值，时间序列采样点具有连续性和数值特性，整个时间序列可以被看做是一个整体的数据对象而非每个单独具体的数值采样点。由于时间序列可以方便的采集得到，大量的时间序列数据可以用来科学研究和深入的数据挖掘。通过时间序列可以更好挖掘、抽取得到可以用于进一步分析和应用的知识，以时间序列数据“形状”上的特性获知每个时间序列的潜在知识。同时将机器学习应用于时间序列，构建基于时间序列数据的机器学习框架。通过原始数据自动的学习得到层次化的特征表示，

特征提取以后，得到了各种各样的特征，但是我们发现得到了特征数据量太大，这样对于那些分类效果不明显的特征就会增加了计算量，影响识别速率。为此我们又做了不少的工作。

### 1.3分类

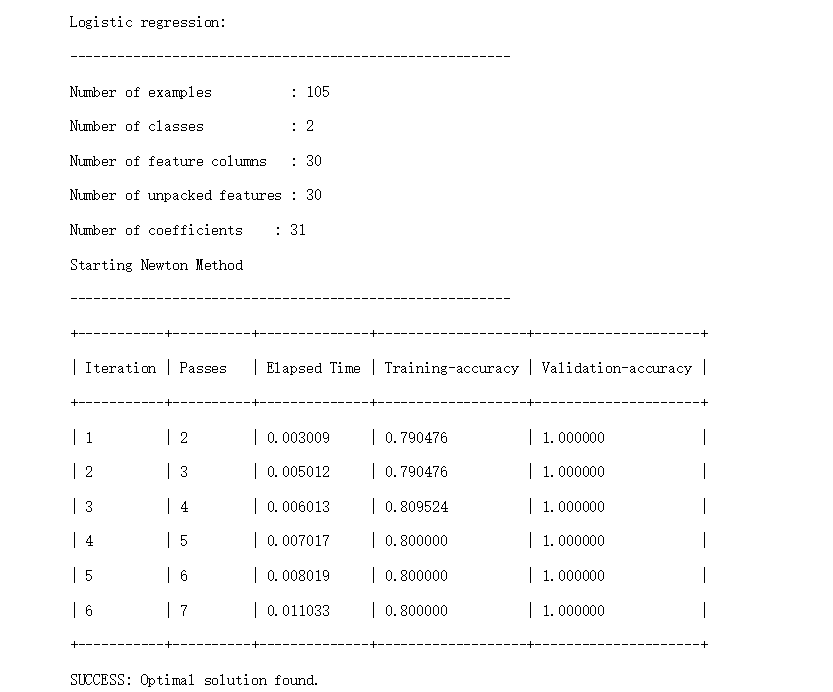


我们运用决策树分类器、支持向量机、逻辑回归等算法，最后从中进行对比，选择最优的。

决策树是数据挖掘中的一种普通的方法。每个决策树可以依靠对源数据库的分割进行数据测试，同时将随机森林分类器将许多决策树结合起来可以提高分类的准确率。

SVM支持向量机属于一般化线性分类器，他们能够同时最小化经验误差与最大化几何边缘区，其模型依赖于无法观测的隐藏变量，对于线性不可分的情况，通过使用非线性映射算法将低维输入空间线性不可分的样本转化为高维特征空间使其线性可分，从而使得高维特征空间采用线性算法对样本的非线性特征进行线性分析成为可能。

逻辑回归算法是分类和预测算法中的一种，通过历史数据的表现对未来结果发生的概率进行预测，使用迭代法对参数进行更新



**逻辑回归效果图**

### 1.4 脑电波获取仪器（Neuro Sky TGAM）



脑电波获取仪器主要由电极片、TGAM脑波模块、电池盒、耳夹、蓝牙从机、LED指示灯等组成。

电极片：可以用来准确检测到极微弱的脑电信号；；

FGAM脑波模块：可以用来测量人体原始脑波信号，它和人体的接触只需要一个干接触点，同时FGAM具有先进的过滤技术，可以排除日常生活环境中的噪音。

耳夹：可以用来过滤掉噪音，具有抗干扰能力。

蓝牙从机：通过主机模块和从机模块同时进行收发，从而实现对数据的传输。

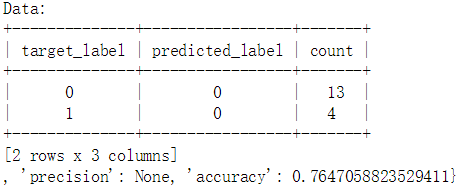
LED指示灯：用来表示蓝牙连接状态。快闪表示未配对，慢闪表示已配对但目前未与从机连接上。长亮表示已经和从机连接上。

### 1.5 软件开发

对了更好的展示癫痫患者的脑电图的数据波形，我们的系统主要是基于labview的脑电波分析预测软件，利用进行图形仿真，在软件界面显示出收集到的比较直观的脑电波波形，并给出预测结果。

### 1.6模型评估图

在测试集上测试结果如下，其中target\_label代表样本标签，predicted\_label代表预测标签,count代表正确预测个数。本次测试结果的精度是0.7647058823529411。



## 2. 软件代码

（1）STM单片机代码

#include "sys.h"

#include "usart.h"

int fputc(int ch, FILE \*f)

{

USART\_SendData(USART1, (uint8\_t) ch);

while (USART\_GetFlagStatus(USART1, USART\_FLAG\_TC) == RESET) {}

return ch;

}

int GetKey (void) {

while (!(USART1->SR & USART\_FLAG\_RXNE));

return ((int)(USART1->DR & 0x1FF));

}

\*/

#if EN\_USART1\_RX //如果使能了接收

//串口1中断服务程序

//注意,读取USARTx->SR能避免莫名其妙的错误

u8 USART\_RX\_BUF[USART\_REC\_LEN]; //接收缓冲,最大USART\_REC\_LEN个字节.

//接收状态

//bit15， 接收完成标志

//bit14， 接收到0x0d

//bit13~0， 接收到的有效字节数目

u16 USART\_RX\_STA=0; //接收状态标记

void uart\_init(u32 bound){

//GPIO端口设置

GPIO\_InitTypeDef GPIO\_InitStructure;

USART\_InitTypeDef USART\_InitStructure;

NVIC\_InitTypeDef NVIC\_InitStructure;

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_USART1|RCC\_APB2Periph\_GPIOA, ENABLE); //使能USART1，GPIOA时钟

//USART1\_TX GPIOA.9

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_9; //PA.9

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_AF\_PP; //复用推挽输出

GPIO\_Init(GPIOA, &GPIO\_InitStructure);//初始化GPIOA.9

//USART1\_RX GPIOA.10初始化

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_10;//PA10

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_IN\_FLOATING;//浮空输入

GPIO\_Init(GPIOA, &GPIO\_InitStructure);//初始化GPIOA.10

//Usart1 NVIC 配置

NVIC\_InitStructure.NVIC\_IRQChannel = USART1\_IRQn;

NVIC\_InitStructure.NVIC\_IRQChannelPreemptionPriority=3 ;//抢占优先级3

NVIC\_InitStructure.NVIC\_IRQChannelSubPriority = 3; //子优先级3

NVIC\_InitStructure.NVIC\_IRQChannelCmd = ENABLE; //IRQ通道使能

NVIC\_Init(&NVIC\_InitStructure); //根据指定的参数初始化VIC寄存器

//USART 初始化设置

USART\_InitStructure.USART\_BaudRate = bound;//串口波特率

USART\_InitStructure.USART\_WordLength = USART\_WordLength\_8b;//字长为8位数据格式

USART\_InitStructure.USART\_StopBits = USART\_StopBits\_1;//一个停止位

USART\_InitStructure.USART\_Parity = USART\_Parity\_No;//无奇偶校验位

USART\_InitStructure.USART\_HardwareFlowControl = USART\_HardwareFlowControl\_None;//无硬件数据流控制

USART\_InitStructure.USART\_Mode = USART\_Mode\_Rx | USART\_Mode\_Tx; //收发模式

USART\_Init(USART1, &USART\_InitStructure); //初始化串口1

USART\_ITConfig(USART1, USART\_IT\_RXNE, ENABLE);//开启串口接受中断

USART\_Cmd(USART1, ENABLE); //使能串口1

}

void USART1\_IRQHandler(void) //串口1中断服务程序

{

u8 Res;

#if SYSTEM\_SUPPORT\_OS //如果SYSTEM\_SUPPORT\_OS为真，则需要支持OS.

OSIntEnter();

#endif

if(USART\_GetITStatus(USART1, USART\_IT\_RXNE) != RESET) //接收中断(接收到的数据必须是0x0d 0x0a结尾)

{

Res =USART\_ReceiveData(USART1); //读取接收到的数据

if((USART\_RX\_STA&0x8000)==0)//接收未完成

{

if(USART\_RX\_STA&0x4000)//接收到了0x0d

{

if(Res!=0x0a)USART\_RX\_STA=0;//接收错误,重新开始

else USART\_RX\_STA|=0x8000; //接收完成了

}

else //还没收到0X0D

{

if(Res==0x0d)USART\_RX\_STA|=0x4000;

else

{

USART\_RX\_BUF[USART\_RX\_STA&0X3FFF]=Res ;

USART\_RX\_STA++;

if(USART\_RX\_STA>(USART\_REC\_LEN-1))USART\_RX\_STA=0;//接收数据错误,重新开始接收

}

}

}

}

#if SYSTEM\_SUPPORT\_OS //如果SYSTEM\_SUPPORT\_OS为真，则需要支持OS.

OSIntExit();

#endif

}

#endif

（2）机器学习模型代码

# -\*- coding:utf-8 -\*-

from scipy.io import loadmat

import matplotlib.pyplot as plt

from matplotlib.pyplot import plot, show

import os

import numpy as np

import pandas as pd

from pandas import DataFrame,Series

from sklearn.decomposition import PCA

import graphlab as gl

data = gl.SFrame.read\_csv(r'F:\Python\project\Epilepsy\patient\_all.csv')

root\_path=r'E:\Data Set\Epilepsy\Patient\_1\Patient\_1'

# root\_path=r'E:\Data Set\Epilepsy\Patient\_2\Patient\_2'

def get\_file\_name(path):

def getListFiles(path):

ret = []

for root, dirs, files in os.walk(path):

for filespath in files:

ret.append(os.path.join(root,filespath))

return ret

ret = getListFiles(root\_path)

del ret[0]

return ret

def mat\_to\_in\_data(path,index):

mat = loadmat(path)

Field\_name = 'interictal\_segment\_' + str(index)

names = mat[Field\_name].dtype.names

# print(names)

ndata = {n: mat[Field\_name][n][0, 0] for n in names}

# print(ndata['data'][0])

return ndata['data'][0:15]

def mat\_to\_pre\_data(path,index):

mat = loadmat(path)

Field\_name = 'preictal\_segment\_'+str(index)

names = mat[Field\_name].dtype.names

# print(names)

ndata = {n: mat[Field\_name][n][0, 0] for n in names}

return ndata['data'][0:15]

def control\_file(file\_name,file\_index):

number = file\_index

if number<=42:

dataFile = file\_name[number]

matdata = mat\_to\_in\_data(dataFile,number)

else:

dataFile = file\_name[number]

matdata = mat\_to\_pre\_data(dataFile,number-42)

return matdata

def data\_PCA(matdata):

date\_rng = pd.date\_range('20170613', periods=3000000, freq='S')

resample\_matric = []

for i in range(15):

ser\_obj = pd.Series(matdata[i], index=date\_rng)

mean\_sample = ser\_obj.resample('100000S').mean()

resample\_matric.append(mean\_sample.values)

data = {'LD\_1': resample\_matric[0],

'LD\_3':resample\_matric[1],

'LD\_4':resample\_matric[2],

'LD\_5':resample\_matric[3],

'LD\_6':resample\_matric[4],

'LD\_7':resample\_matric[5],

'LD\_8':resample\_matric[6],

'RD\_1':resample\_matric[7],

'RD\_2':resample\_matric[8],

'RD\_3':resample\_matric[9],

'RD\_4':resample\_matric[10],

'RD\_5':resample\_matric[11],

'RD\_6':resample\_matric[12],

'RD\_7':resample\_matric[13],

'RD\_8':resample\_matric[14],}

frame = DataFrame(data)

np\_data = np.array(frame)

pca=PCA(n\_components=1)

pca\_data=pca.fit\_transform(frame)

return pca\_data.transpose()

file\_name = get\_file\_name(root\_path)

pd\_data = []

for i in range(1,61):

np\_pca = data\_PCA(control\_file(file\_name, i))

pd\_data.append(np\_pca[0])

pd\_data = np.array(pd\_data)

df = DataFrame(pd\_data)

def getListFiles(path):

ret = []

for root, dirs, files in os.walk(path):

ret.append(files)

del ret[0][0]

del ret[0][0]

return ret

root\_path=r'E:\Data Set\Epilepsy\Patient\_2\Patient\_2'

df\_name = getListFiles(root\_path)

df\_Column\_name = []

for name in df\_name[0]:

df\_Column\_name.append(name.split('.')[0])

name\_of\_df = []

for i in range(0,60):

name\_of\_df.append(df\_Column\_name[i])

df\_Column\_name = name\_of\_df

y = []

for i in range(1,61):

if i<=42:

y.append(0)

else:

y.append(1)

print(len(y))

df['y'] = y

df['name'] = df\_Column\_name

train\_data,test\_data = data.random\_split(.8,seed=0)

model = gl.classifier.create(train\_data, target='y',

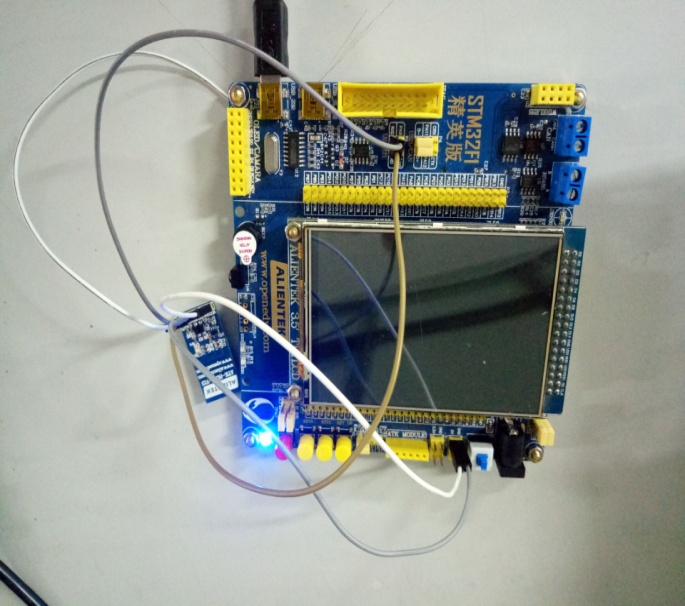
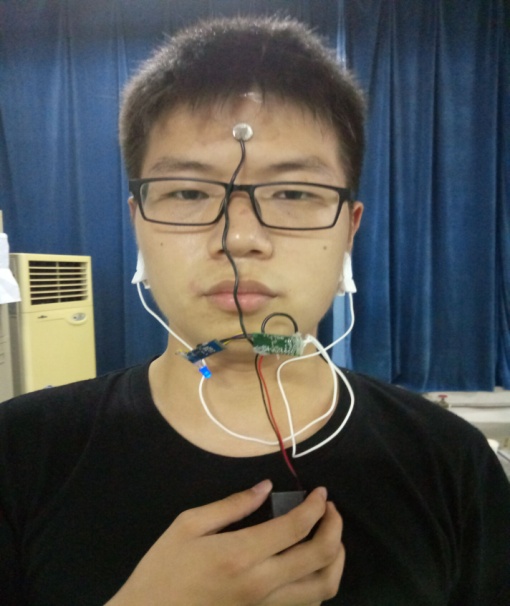
features=['a'+ str(i) for i in range(0,30)])

print(model.evaluate(test\_data))

## 3. 实物照片

（1）如图一所示为脑电波提取仪器，分为电源，蓝牙发送模块，脑电波感应电极，耳夹（零点电极模块），TGAM脑电波模块。此仪器起到实时收集脑电波并通过蓝牙模块将脑电波发送到中转模块的作用。

（2）如图二所示为脑电波信号接收中转模块，由STM32处理模块和蓝牙接收模块构成，将接收到的脑电波信号进行相关的算法处理，可以通过串口传输到电脑软件。



图一 脑电波提取仪器 图二 STM32单片机模块

## 4. 创新点

（1）以机器学习为核心算法，对EEG数据进行特征提取、分类，实时获取患者脑部状态，在发病前得到精确预测，该系统有较高的准确性和预测效果。

（2）通过STM32单片机，结合蓝牙模块将数据传输到上位机LABVIEW中。

（3）基于机器学习的脑电波处理平台。采用scikit-learn,tsfresh，pandas完成特征工程，通过构建随机森林，支持向量机，逻辑回归等混合多样化模型实现最终平台的算法架构。

（4）软硬件结合，方便携带和检测。

（5）基于LABVIEW的脑电波分析展示软件，通过机器学习，PCA，傅里叶变换等算法将采集到的脑电波进行区分判断，在软件界面显示出收集到的脑电波波形图，并给出预测结果。

## 5. 实用性

癫痫病预测系统是基于机器学习的利用电极和蓝牙模块获取病人脑电波，通过串口将采集到的数据传输到电脑，用上位机显示波形，另外用人工智能机器学习模型预测癫痫病发作概率，为采取治疗提供信号。

智能机器学习使癫痫病预测发作概率更加准确，而随着我们技术的一步步成熟，最终必将发展成为可穿戴的设备。

## 6. 市场应用前景

目前癫痫人口约占全世界人口的1%左右，我国的现有癫痫病人约有1000多万人，每年的发病率们的一大困扰。对于我们这一领域的发展，无疑给癫痫患者的健康再次提供了一个全新的预测方向，很好的避免了癫痫患者因为对治疗不得要领而给癫痫患者及其家人带来较大的经济负担和巨大的精神心理负担。同时对于那些2-3年以上完全无发作，可以考虑停药，但在发作前需要进行一次EEG检查，我们癫痫病预测系统可以精确的检测放电，确定患者是否有进一步要发病的可能。

目前，脑电波相关的技术大多应用于医疗领域，特别是对于癫痫疾病等脑部疾病的病人，但是一方面由于人的大脑过于复杂，人类对它的研究还比较初级，另一方面缺乏足够长的时间记录以及足够多的数据以便研究人员可以运用算法，使癫痫发作预测一直受到阻碍，大多数科研机构仍然停留于实验室阶段。我们的癫痫预测系统通过电极检测到脑部信号，运用机器学习，已经可以初步对癫痫患者进行发病前预测发病概率，在未来进行相关推广还是具有很大的优势。