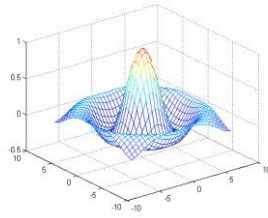


五一数学建模竞赛



题目：煤矿深部开采冲击地压危险预测模型的分析与研究

关键词：傅里叶变换、随机森林算法、朴素贝叶斯、Platt 缩放

摘要：煤炭是中国的主要能源和重要的工业原料。在冲击地压已成为威胁中国煤矿安全生产最重要的灾害之一的当下，实现冲击地压的监测预警和有效防控对于煤矿安全生产来说具有重要意义。本文通过建立评价模型、分类模型，运用机器学习方法研究干扰信号、前兆特征信号的趋势特征与预测方法，并给出预警前兆信号概率的具体方法。

针对问题一：我们首先对附件 1 中数据进行预处理，未发现缺失数据，进而分别绘制 EMR 和 AE 中 C 类干扰信号的时域强度图。为了提取出图中 EMR 和 AE 干扰信号的特征，我们选取均值、方差、峰值、峰值因子、偏度、主要频率、频谱能量密度七项指标作为评判干扰信号的依据。针对主要频率、频谱能量密度，我们采用**傅里叶变换**从频域进行分析。根据附件 1 数据计算这七项指标，我们对附件 2 中的三个时间段的 EMR 和 AE 强度数据进行识别，采用**滑动窗口**进行局部特征提取，并与七项指标依次比较，筛选出与指标相对误差在 10% 以内的数据，视为干扰信号，并提取出前五个干扰信号区间填入表 1 和表 2 中。

对于问题二：我们在对附件 1 表单中电磁辐射和声发射前兆特征信号数据进行提取后，通过计算其时域、频域的具体统计特征，结合特征可视化图表与正常信号的直观对比得到前兆特征信号数据的趋势特征。随后我们依据趋势特征提取对应统计量在附件 1 表单基础上构建前兆特征信号分类的训练集与测试集，训练**随机森林分类模型**。在测试集上发现电磁辐射前兆特征信号分类准确率为 97.99%；声发射前兆特征信号分类准确率为 99.40%，**验证了分类模型的合理性**。最后依据分类模型对题设要求的所有电磁辐射和声发射信号进行前兆特征信号标签的二分类，将连续出现的标记为前兆特征信号的时间节点进行聚合，得到所有前兆特征信号所在的时间区间。

对于问题三：我们选定朴素贝叶斯模型作为我们的分类器，对数据进行了**小波降噪**和 **Platt 缩放**处理，在保留数据本质的同时，优化其特征表示，以提高模型的分类准确性和泛化能力。在得出结论后，我们使用**学习曲线**和 **ROC 曲线**来评估模型的准确率。学习曲线我们使用的朴素贝叶斯模型在训练过程中的学习进展。ROC 曲线则能够展示模型在不同阈值下的性能表现，从而更全面地评估其分类能力。

目录

1.问题重述	3
2.符号说明	3
3.模型假设	3
4.问题分析	4
4.1 问题一分析	4
4.2 问题二分析	4
4.3 问题三分析	4
5.模型建立与求解	5
5.1 扰信号数据的特征分析和识别方法	5
5.1.1 干扰信号特征提取	6
5.1.2 干扰信号的识别	8
5.2 前兆特征信号的趋势特征分析和信号识别方法	8
5.2.1 分析电磁辐射和声发射信号中前兆特征信号的变化趋势	8
5.2.2 前兆特征信号分类模型的建立	10
5.2.3 模型结论	12
5.3 基于朴素贝叶斯模型的前兆特征发生概率预测	14
5.3.1 数据处理	14
5.3.2 朴素贝叶斯模型的建立	17
5.3.3 模型检验	17
5.3.4 模型结论	18
6.模型优缺点与进一步讨论	18
7.参考文献	19
附录	20

一、问题重述

冲击地压事故已成为各类深部煤岩动力灾害事故中威胁中国煤矿安全生产最重要的灾害之一，监测电磁辐射(EMR)信号和声发射(AE)信号数据可通过其变化趋势判断目前工作面或巷道是否存在冲击地压危险。本体要求建立数学模型，依据附件的相关数据，解决如下三个问题：

- （1）分析电磁辐射和声发射信号中干扰信号数据的特征；结合特征对题设目标信号时间段中的干扰信号时间段进行识别预测；列出最早出现的 5 个干扰信号时间段。
- （2）分析电磁辐射和声发射信号中前兆特征信号的变化趋势特征；结合特征对题设目标信号时间段中的前兆特征信号时间段进行识别预测；列出最早出现的 5 个前兆特征信号时间段。
- （3）根据题设非连续时间段的数据，结合第（1）问中的数据清洗方法以及第（2）问中的变化趋势特征预测每个时间段最后时刻出现前兆特征数据的概率。

二、符号说明

关键符号	符号说明
rolling_mean	滑动窗口均值
rolling_std	滑动窗口标准差
fft_mean	傅里叶变换频率分量均值
fft_max	傅里叶变换频率分量最大值
rolling_var	滑动窗口方差
d1	一阶差分
fft_std	傅里叶变换频率分量标准差

三、模型假设

- 1.假设附件中对电磁辐射(EMR)信号和声发射(AE)信号的测量值与类别标记是准确无误的，不存在测量偏差；
- 2.略其他信号因素与信号类别的关联和影响；
- 3.假设干扰信号与前兆特征信号均有明显区别于正常信号的特征存在。

四、问题分析

4.1 问题一的分析

问题一第一问要求应用附件 1 的数据给出干扰信号的特征，然后运用到第二问对附件 2 的数据进行识别并给出三个时间段最早发生的五个干扰信号区间。首先我们抽取出附件 1 中的干扰信号数据，画出时域上信号强度的图像，先通过观察初步确定信号的特征，再通过计算数据的数字特征，如均值，方差，峰值，峰值因子，偏度，又通过对离散点的傅里叶变换在频域求得主要频率和频谱能量密度。接着设计了一个窗口大小为 100 的滑动窗口，对附件 2 中的数据进行逐步识别，通过设置允许误

差来筛选在上述指标范围波动 10%以内的数据并将其认定为是干扰信号。

4.2 问题二的分析

问题二要求分析电磁辐射和声发射信号中的前兆特征信号的变化趋势；给出数据的趋势特征；并给出具体的预测方法对题设时间区间中前兆特征信号进行识别；给出最早发生的 5 个前兆特征信号所在的时间区间。我们首先对表单中电磁辐射和声发射前兆特征信号数据进行提取，通过计算其时域、频域的具体统计特征，结合特征可视化图像直观得到前兆特征信号数据的趋势特征。接着建立随机森林分类模型对整体数据进行拟合，借助混淆矩阵等模型评测标准对分类模型进行检验验证模型是否合理。最后通过在题设时间区间的数据进行前兆特征数据标签的预测，将连续出现的标记为前兆特征信号的时间节点进行聚合，得到所有前兆特征信号所在的时间区间。

4.3 问题三的分析

对于问题三，我们选定朴素贝叶斯模型作为我们的分类器，对数据进行了小波降噪和 Platt 缩放处理，在保留数据本质的同时，优化其特征表示，以提高模型的分类准确性和泛化能力。在得出结论后，我们使用学习曲线和 ROC 曲线来评估模型的准确率。学习曲线我们使用的朴素贝叶斯模型在训练过程中的学习进展。ROC 曲线则能够展示模型在不同阈值下的性能表现，从而更全面地评估其分类能力。

五、模型的建立与求解

5.1 问题一：干扰信号数据的特征分析和识别方法

5.1.1 干扰信号特征提取

我们首先对附件 1 中 EMR 和 AE 干扰信号进行提取，做出以下两图：

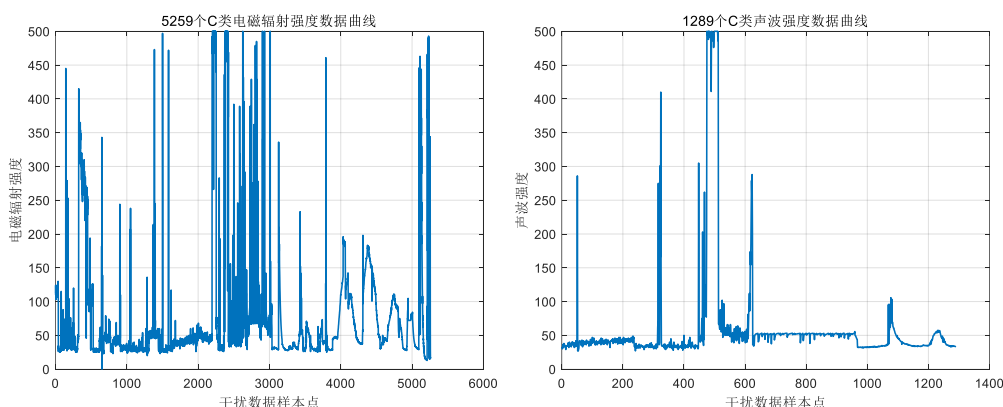


图 1：EMR 干扰信号(左)和 AE 干扰信号(右)

接着我们选取了七个指标：均值、方差、峰值、峰值因子、偏度、主要频率、频谱能量密度来从时域和频域上对干扰信号做出特征提取。

a. 时域特征：

均值：均值是数据集的中心趋势的度量。设有随机变量 $x_1, x_2, \dots, x_n$ ，则它们的均值可表示为：

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

方差：方差是衡量数据的离散程度。较大的方差表示数据点分布不稳定，而较小的方差表示数据更接近均值。其定义为：

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

峰值/峰值因子：峰值是数据局域范围内的极大值，峰值因子是数据的最高值与平均值之比，即

$$f = \frac{\max\{x_i\} - \min\{x_i\}}{\bar{x}}$$

较高的峰值因子表示信号波动性较大。

数据的峰度：是描述数据分布尖峰程度的统计量。峰度衡量了数据分布相对于正态分布的尖峰或平缓程度。具体来说，峰度反映了数据分布的尾部厚度和尖峰程度。峰度的取值范围为 $[1, +\infty)$ ，完全服从正态分布的数据的峰度值为 **3**，峰度值越大，概率分布图越高尖，峰度值越小，越矮胖。

数学上，数据的峰度可以通过以下公式计算：

$$Kurtosis = \frac{1}{n} \sum_{i=1}^n \left(\frac{E(X_i - \mu)}{\sigma} \right)^4 - 3$$

数据的偏度：是描述数据偏离正态的分布的程度。偏度的取值范围为 $(-\infty, +\infty)$ ，当偏度 >0 时，概率分布图右偏；当偏度 <0 时，概率分布图左偏；当偏度 $=0$ 时表示数据均匀地分布在均值两侧。其定义式如下：

$$\frac{1}{n} \sum_{i=1}^n \left(\frac{E(X_i - \mu)}{\sigma} \right)^3$$

b. 频域特征

频域特征是描述不同信号的数据在不同频率下的贡献，可以清楚地反映哪些频率的信号对总输出信号的贡献最大。频域特征包括频谱能量，频谱带宽，幅值和相位等。我们通过**傅里叶变换(Fourier Transformation)**将处于时域的干扰信号变换到频域，进而能够得到在时域无法得到的某些特征。傅里叶变换的定义式如下：

$$F[f(t)] = \int_{-\infty}^{+\infty} e^{-i\omega t} f(t) dt$$

如果信号是连续的，就可以采用连续傅里叶变换(CFT,如上所述)，但本题的信号是离散数据，可以采用离散傅里叶变换(DFT)：

$$F[f(t)] = \sum_{n=0}^{N-1} e^{-i \frac{n\omega}{N}} f(n) \quad , \quad n = 0, 1, 2, \dots, N-1$$

其中，N 是信号长度， $f(n)$ 是离散的信号在时域上的表示。通过傅里叶变换我们可以得到频谱图，通过绘制的频谱图，可以找到数据的一些特征信息：

主要频率成分：通过观察频谱图的峰值位置，可以找到数据中振幅最大值对应的主要的频率，它们代表了频域上对时域信号贡献最大的那部分信号。

频谱能量：频谱能量是所有频率成分的信号振幅的平方和：

$$E = \sum_{n=0}^{N-1} |x(n)|^2$$

频谱能量密度：描述了在一定区间范围内频谱的平均能量，即频谱能量除以区间长度，反映了数据在某一频率区间下的能量分布情况。

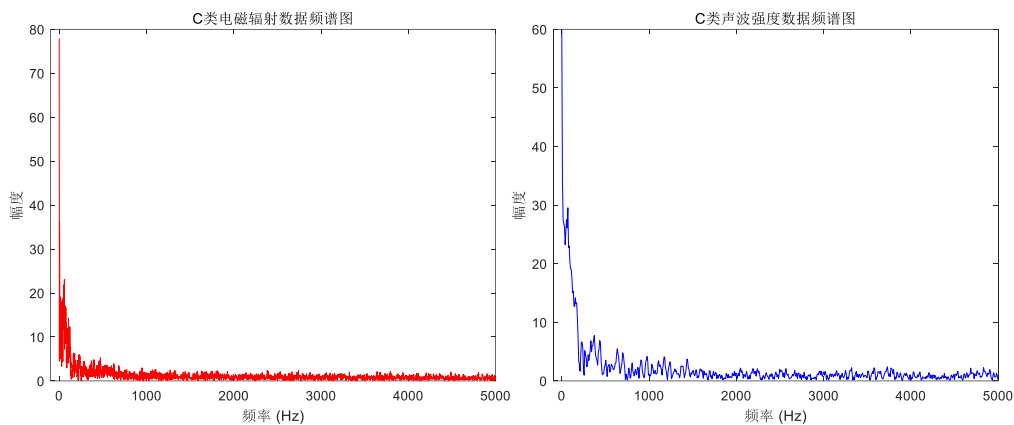


图 2：EMR (左) 和 AE (右) 干扰信号频谱图

综上，我们通过计算得到 EMR 和 AE 干扰信号的七项指标如下表所示：

表 1：EMR 和 AE 七项指标对应数据

	均值	方差	峰值	峰值因子	偏度	主要频率成分	频谱能量密度
EMR	77.9585	8229.3701	12.9982	6.4137	3.1238	0.00019015	14305.3337
AE	59.9691	6467.6703	26.0055	7.8874	4.8817	0.0007758	58364.4296

由表中数据可知，AE 数据的峰值因子高于 EMR 数据的峰值因子，这表明 AE 信号的波动性更高。EMR 数据的方差为 8229.37，AE 数据的方差为 6467.67。因此，EMR 数据的离散程度略高于 AE 数据。EMR 数据的均值为 77.96，AE 数据的均值为 59.97。这表明，在观察期间，EMR 数据的均值略高于 AE 数据的均值。EMR 的主要频率成分比 AE 要低，说明 EMR 主要由相对低频的信号贡献，而 EMR 高频信号相对于 AE 贡献较少。从图 2 的观察中也可以印证这一点。

5.1.2 干扰信号的识别

这里我们采用滑动窗口对数据进行识别，其原理是在一串数据上通过滑动窗口来计算窗口内数据的特征，进而与已经得到的标准值进行比较，从而筛选出符合特征的数据。这里我们采用窗口大小为 100 的窗口，然后设置允许误差为 10%，即如果窗口内的数据特征与第一问得到的特征的相对误差在 10%以内则视为干扰信号。

通过计算,我们得到三个时间段的 EMR 和 AE 前五个干扰信号发生的区间如下表所示:

表 1-1 电磁辐射时间段 1 干扰信号时间区间

序号	时间区间起点	时间区间终点
1	2022-5-18 21:49:51	2022-5-18 22:04:27
2	2022-5-18 23:45:03	2022-5-19 0:16:08
3	2022-5-19 1:53:05	2022-5-19 2:37:00
4	2022-5-19 3:51:59	2022-5-19 8:15:21
5	2022-5-19 9:57:48	2022-5-19 10:39:53

表 1-2 电磁辐射时间段 2 干扰信号时间区间

序号	时间区间起点	时间区间终点
1	2020-4-14 7:03:31	2020-4-14 9:35:30
2	2020-4-15 2:52:36	2020-4-15 4:06:48
3	2020-4-15 19:33:06	2020-4-15 20:12:47
4	2020-4-15 21:38:57	2020-4-15 22:12:50
5	2020-4-15 23:22:32	2020-4-16 0:15:46

表 1-3 电磁辐射时间段 3 干扰信号时间区间

序号	时间区间起点	时间区间终点
1	2021-11-22 1:48:48	2021-11-22 2:50:25
2	2021-11-22 4:31:53	2021-11-22 5:40:44
3	2021-11-22 10:34:18	2021-11-22 11:35:53
4	2021-11-22 16:38:33	2021-11-22 17:38:21
5	2021-11-22 20:25:03	2021-11-22 21:28:27

表 2-1 声发射时间段 1 干扰信号时间区间

序号	时间区间起点	时间区间终点
1	2022-4-2 21:38:12	2022-4-2 22:40:23
2	2022-4-3 15:20:51	2022-4-3 16:23:01

3	2022-4-3 16:08:24	2022-4-3 17:08:45
4	2022-4-3 17:03:16	2022-4-3 18:03:39
5	2022-4-3 17:58:09	2022-4-3 19:00:20

表 2-2 声发射时间段 2 干扰信号时间区间

序号	时间区间起点	时间区间终点
1	2022-10-10 6:12:56	2022-10-10 7:11:28
2	2022-10-12 23:41:32	2022-10-13 0:42:01
3	2022-10-19 0:16:54	2022-10-19 1:17:20
4	2022-10-21 22:23:36	2022-10-21 23:23:58
5	2022-10-23 15:33:32	2022-10-23 16:33:54

表 2-3 声发射时间段 3 干扰信号时间区间

序号	时间区间起点	时间区间终点
1	2021-11-6 17:14:56	2021-11-6 18:14:42
2	2021-11-6 17:23:59	2021-11-6 18:23:46
3	2021-11-6 18:54:34	2021-11-6 19:54:21
4	2021-11-6 21:55:44	2021-11-6 22:55:31
5	2021-11-7 0:56:53	2021-11-7 1:56:42

5.2 问题二：前兆特征信号的趋势特征分析和信号识别方法

本问分为两个小问：（1）分析电磁辐射和声发射信号中的前兆特征信号的变化趋势；（2）给出具体的预测方法识别前兆特征信号；（3）分析分类模型的合理性。我们通过前兆特征信号的可视化图像分析直观总结前兆特征信号的变化趋势特征。接着建立随机森林分类模型对整体数据进行拟合，验证模型是否合理。最后预测题设时间段内时间节点的前兆特征信号标签，将连续标记为前兆特征信号的时间节点进行聚合获取前兆特征信号出现的时间区间。

5.2.1 分析电磁辐射和声发射信号中前兆特征信号的变化趋势

依据所给附件，要研究电磁辐射与声发射信号中前兆特征信号的变化趋势，我们可以采用图像直观分析的方法，对变化趋势进行分析。

● 利用图像初步分析前兆特征信号变化趋势

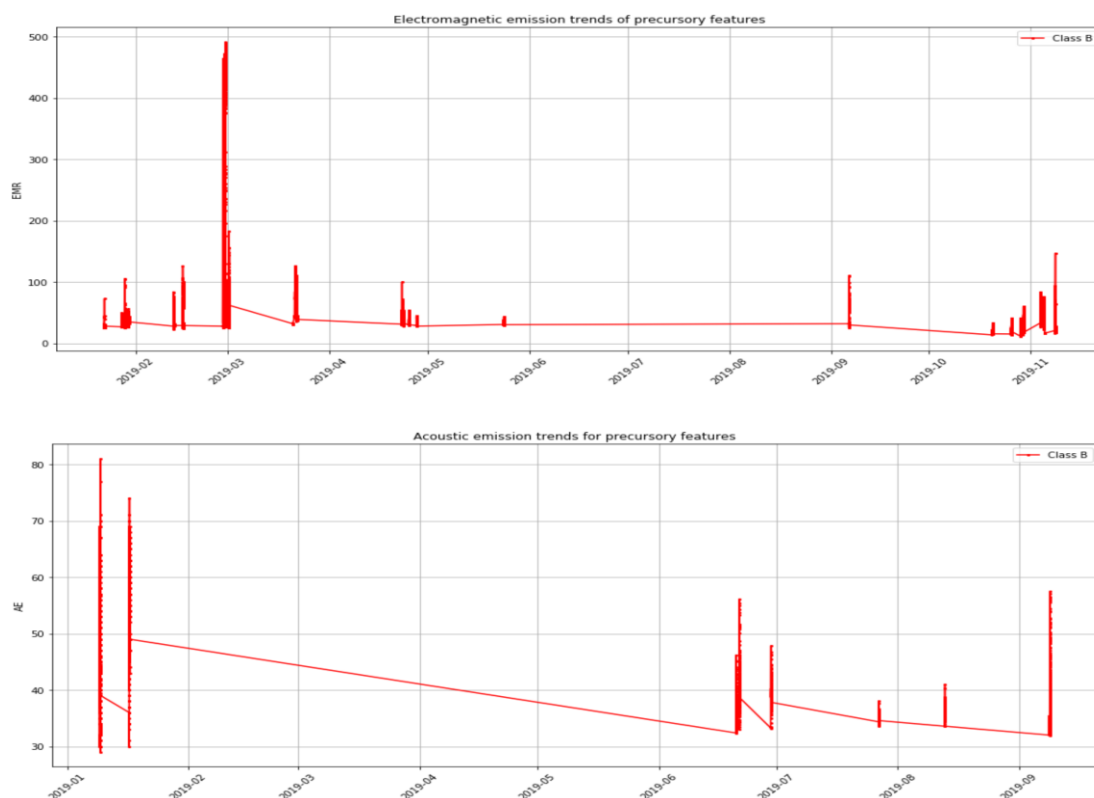


图 3：电磁辐射（上）与声发射信号（下）的前兆特征数据趋势

观察上述图像,我们初步得到电磁发射与声发射信号的前兆特征信号趋势特征。

突然增长与异常峰值：在周期性波动中，观察到某些时期信号呈现出相对稳定的平稳区间，然后突然增长，具体表现为变化率的短期急剧波动与较高的峰值体现。

● 滑动窗口特征提取

附件给出的数据为明确的时间序列数据，且各时间点数据特征并不单独存在，而是与每个时间点的前后数据互相关联。因此，特征提取需要充分考虑段时间内的所有时间节点，所以我们选择使用滑动窗口的方法进行特征提取。

我们首先将窗口大小设为 30,针对附件 1 数据首先分别筛选出 EMR 信号与 AE 信号的前兆特征数据（B 类数据），然后仅对这些数据滑动计算其均值、标准差这两个指标，特征提取如图 4 所示：

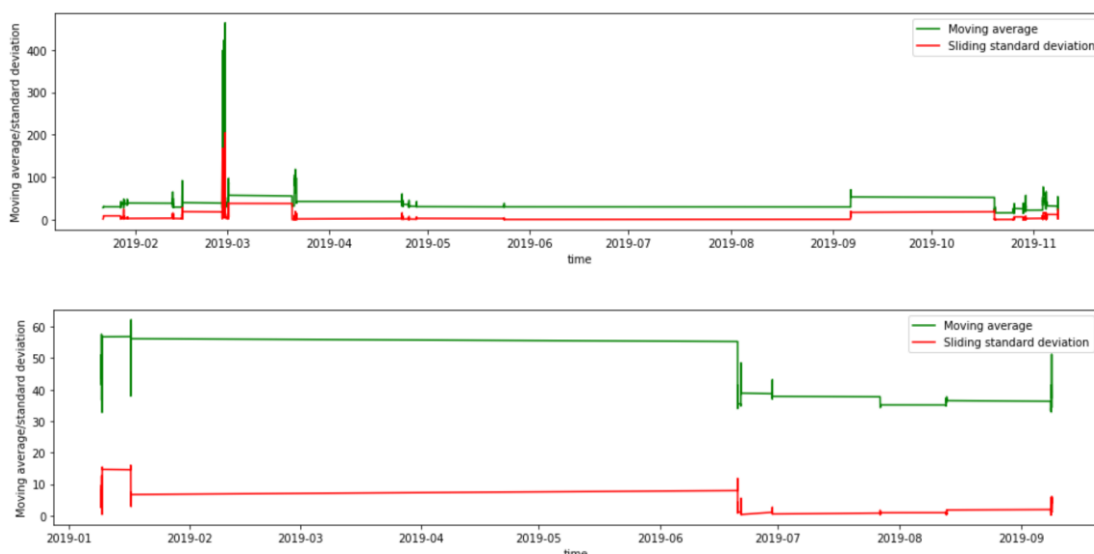


图 4：电磁辐射（上）与声发射信号（下）的滑动窗口特征

观察上述图像，我们得到电磁发射与声发射信号的前兆特征信号趋势特征。

滑动均值与标准偏差的异常波动：利用滑动均值可将原始数据进一步平滑化，观察到前兆信号发生时期滑动均值与标准偏差均出现了较大规模的突然性增加。

● 频域特征提取

对于时间序列信号，频域分析可将时域信号转换为频域表示，从中提取频域特征。这些特征有助于我们更好地理解信号的频率成分和周期性，揭示信号中的重要信息。离散时域信号的有效频域分析方法为离散傅里叶变换（Discrete Fourier Transform, DFT），但在较大的数据集上，我们更倾向于采用快速傅里叶变换（Fast Fourier Transform, FFT）方法迅速获得信号的频域表示。

仅对 EMR 信号与 AE 信号的前兆特征数据进行快速傅里叶变换，提取其均值、标准差与最大值三项统计特征，特征提取如图 5 所示：

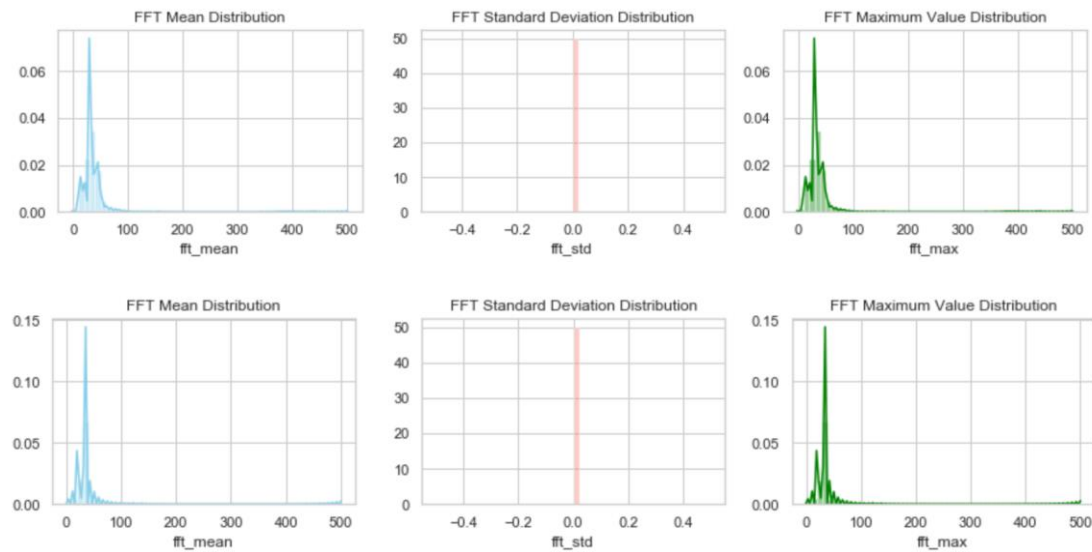


图 5：电磁发射（上）与声发射信号（下）的傅里叶统计特征

观察上述图像，我们得到电磁发射与声发射信号的前兆特征信号趋势特征。

傅里叶特征的聚集趋势：对于前兆特征信号进行时域-频域转换即离散傅里叶变换后，可以发现其傅里叶变换均值与最大值特征呈现一段范围内的频率聚集。

综上所述，我们可以得出 EMR 和 AE 信号危险发生前（前兆特征）数据的趋势特征：

- 1) 突然增长与异常峰值：在周期性波动中，观察到某些时期信号呈现出相对稳定的平稳区间，然后突然增长，具体表现为变化率的短期急剧波动与较高的峰值体现；
- 2) 滑动均值与标准偏差的异常波动：利用滑动均值可将原始数据进一步平滑化，观察到前兆信号发生时期滑动均值与标准偏差均出现了较大规模的突然性增加；
- 3) 傅里叶特征的聚集趋势：对于前兆特征信号进行时域-频域转换即离散傅里叶变换后，可以发现其傅里叶变换均值与最大值特征呈现一段范围内的频率聚集。

5.2.2 前兆特征信号分类模型的建立

根据已有趋势特征，主要依据趋势特征中所体现的统计特征对整体数据进行特征矩阵的建立，进而可以依据特征矩阵作为训练样本建立 EMR 和 AE 信号中前兆特征信号的机器学习分类模型。

● 特征矩阵的确立

借助已经得到的趋势特征，我们对附件 1 中所有数据计算其一阶差分、滑动均值/方差/标准差、傅里叶变换频率分量均值/标准差/最大值七个指标，并将所有前兆特征数据（B 类数据）的标签记为 1，得到全数据集的特征矩阵，以声波强度（AE）为例：

	声波强度 (AE)	时间 (time)	类别 (class)	fft_mean	fft_std	fft_max	d1	rolling_mean	rolling_var	rolling_std	前兆信号
0	58.000	2019-01-09 00:00:17	D/E	58.000	0.0	58.000	0.000	44.8000	23.600000	2.936362	0
1	43.000	2019-01-09 00:00:48	D/E	43.000	0.0	43.000	-15.000	44.8000	112.500000	2.936362	0
2	42.000	2019-01-09 00:01:19	D/E	42.000	0.0	42.000	-1.000	44.8000	80.333333	2.936362	0

● 数据过采样处理

我们在对数据进行前兆特征数据（B 类数据）的筛选之后发现在整个训练集上样本标签并不均衡，即标签为 0 的样本较多而标签为 1 的样本较少，这将导致模型对标签为 1 的数据特征学习不充分，进而失去良好的泛化效果。

过采样是一种用于处理类别不平衡数据的技术，通过增加少数类样本的复制或合成新样本的方式，来提高少数类样本在数据集中的比例。这通常用于解决在机器学习任务中因类别不平衡而导致模型性能下降的情况。

SMOTE（Synthetic Minority Over-sampling Technique）算法是一种用于处理类别不平衡数据集的过采样方法。它通过在特征空间中合成新的少数类样本，从而增加少数类样本的数量，达到平衡样本分布的目的。

SMOTE 算法的基本思想是：对于少数类样本中的每个样本，从它的最近邻中随机选择一个样本，在该样本和原样本之间的连线上随机生成新的样本。这样可以在特征空间中扩大少数类样本的分布，从而缓解类别不平衡问题。

借助 imbalanced-learn 科学库，我们可以快速地对数据集利用 SMOTE 算法进行有效的过采样处理，经过处理的数据集两类数据的数据量相等，成为适合机器学习的良好数据集。

● 随机森林分类模型的建立

本问的实际要求可重述为判断附件 2 中的数据是否为前兆特征数据，进而简化为二分类问题。本题所给出的训练集具有样本空间大，数据具有不平衡性等特征；随机森林算法可以整合多个决策树的预测结果，通过集成学习的方式平衡各类别的重要性，通过投票机制来降低模型方差，从而减轻决策树算法固有的过拟合问题，提供清晰的分类标准，所以本文采用随机森林算法进行分类器的构建。

随机森林算法的核心思想是“集成学习”^[2]，即将固定数量和不同特征的决策树集成起来，使其具有较高的预测性能和鲁棒性。随机森林算法的优点在于它对数据集的扰动具有强大的适应性，在许多方式中对不良事实具有较高的鲁棒性。

假设有 N 个样本， M 个特征可以进行分类任务，每个样本都拥有相应的标记。然后，建立 T 颗决策树，其中每棵树都是在一个样本子集上随机挑选的，并且在构建每棵树的过程中，只考虑了所有特征的一部分。每一个特征被称为一个自变量，所有自变量可以表示为 $\{X_1, X_2, \dots, X_m\}$ 。

对于每一棵树，随机森林算法定义了以下步骤：

Step 1: 从样本集中通过有放回抽样方式抽取训练集样本；

Step 2: 从所有自变量中通过无放回抽样方式随机抽取 k 个自变量，其中 k 远小于 m 。抽取的自变量集合可以表示为 $\{f_1, f_2, \dots, f_k\}$ 。

Step 3: 基于训练样本和生成的自变量子集构建一棵决策树，使用某种标准衡量特征的重要性，以确定在树中选择第一个分裂节点的特征。

Step 4: 重复 Step1-3 操作 T 次，生成 T 棵决策树。

通过调整决策树的数量、权重、资源分配，随机森林最终依据所有决策树的分类结果投票决策出样本的标签类别。

● 随机森林分类模型的求解

本文通过 Python，依靠 scikit-learn 科学库进行随机森林模型的构建、拟合、求解与评价。我们划分全数据集为训练集:测试集为 7:3 的比例，即 70%的数据用于随机森林模型权重的训练，另外 30%的数据用于评估训练完成的随机森林模型是否可以良好地对前兆特征信号进行预测分类。

EMR 与 AE 信号前兆特征数据的随机森林分类报告如下图所示：

Accuracy: 0.9798713235294118					Accuracy: 0.9940899249896122				
Classification Report:					Classification Report:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.99	0.97	0.98	179110	0	1.00	0.99	0.99	182522
1	0.97	0.99	0.98	179930	1	0.99	1.00	0.99	183294
accuracy			0.98	359040	accuracy			0.99	365816
macro avg	0.98	0.98	0.98	359040	macro avg	0.99	0.99	0.99	365816
weighted avg	0.98	0.98	0.98	359040	weighted avg	0.99	0.99	0.99	365816

图 6：电磁辐射（左）与声发射（右）前兆特征信号的随机森林分类报告

EMR 与 AE 信号前兆特征数据的随机森林混淆矩阵可视化如下图所示：

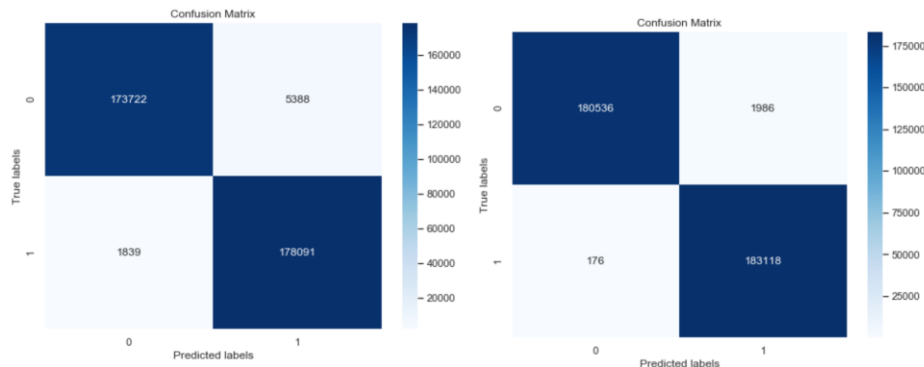


图 7：电磁辐射（左）与声发射（右）前兆特征信号的随机森林分类混淆矩阵

EMR 与 AE 信号前兆特征数据的随机森林分类 ROC 曲线如下图所示：

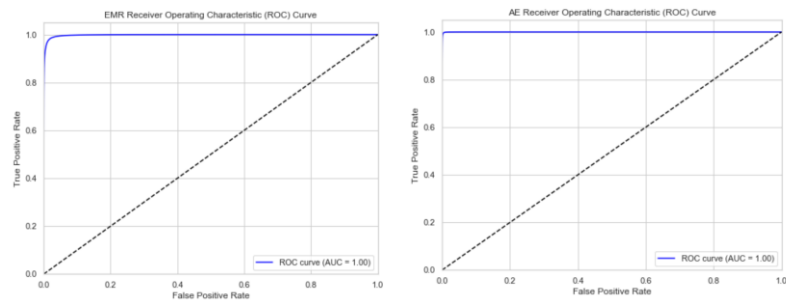


图 8：电磁辐射（左）与声发射（右）前兆特征信号的随机森林分类 ROC 曲线

混淆矩阵（Confusion Matrix）是在机器学习领域中用来评估分类模型性能的一种表格，用于展示模型在不同类别上的预测结果。通过混淆矩阵，我们可以计算出一些评估分类模型性能的指标，例如准确率（Accuracy）、精确率（Precision）、召回率（Recall）、F1 分数（F1-score）等。

ROC 曲线是一种用于评估分类器二元分类性能的常用工具。ROC 曲线通常以假正例率（False Positive Rate, FPR）为横轴，真正例率（True Positive Rate, TPR，

也称为召回率)为纵轴。在 ROC 曲线中,我们可以观察到不同阈值下的分类器性能表现。一般情况下,ROC 曲线越靠近左上角,表示分类器的预测性能越好。ROC 曲线下方的面积(AUC)可以用来量化分类器的性能。AUC 的取值范围在 0.5(随机猜测)到 1 之间,数值越大表示分类器性能越好。

集成主要评价指标,制表如下:

评价指标	准确率	精度	召回率	F1 分数	AUC
数值	97.99%	98.00%	98.00%	0.98	1.00

评价指标	准确率	精度	召回率	F1 分数	AUC
数值	99.40%	99.00%	99.00%	0.99	1.00

表 2: 电磁辐射(上)与声发射(下)前兆特征信号的主要评价指标

通过以上指标,我们可以评判随机森林分类模型在全数据集上已经取得了较好的分类效果,适于进行未见数据的预测工作。

5.2.3 模型结论

在附件 2 所提供的数据集上,我们首先筛选出题设时间区间的所有数据,并以相同的特征构造方法构造特征矩阵,分别在三个时间区间上进行预测得到每个时间节点的前兆特征数据标签,聚合所有连续的时间节点得到前兆特征信号所在的时间区间,填入题设表格中:

表 3-1 时间段 1 电磁辐射前兆特征时间区间

序号	时间区间起点	时间区间终点
1	2020/4/8 0:44:53	2020/4/8 0:46:49
2	2020/4/8 0:49:44	2020/4/8 0:55:32
3	2020/4/8 1:09:06	2020/4/8 1:11:02
4	2020/4/8 1:45:38	2020/4/8 1:57:15
5	2020/4/8 2:00:10	2020/4/8 2:06:56

表 3-2 时间段 2 电磁辐射前兆特征时间区间

序号	时间区间起点	时间区间终点
1	2021/12/19 9:24:43	2021/12/19 9:31:58
2	2021/12/19 9:35:35	2021/12/19 9:44:39
3		
4		
5		

表 4 声发射前兆特征时间区间

序号	时间区间起点	时间区间终点
1	2021/11/26 5:14:26	2021/11/26 5:19:53
2	2021/11/26 5:23:30	2021/11/26 5:36:11
3	2021/11/26 5:56:07	2021/11/26 6:08:48
4	2021/11/27 2:37:33	2021/11/27 2:46:36
5	2021/11/27 2:57:29	2021/11/27 2:59:17

5.3 问题三：基于朴素贝叶斯模型的前兆特征发生概率预测

5.3.1 数据处理

● 使用小波分析降噪处理

噪声指的是与我们感兴趣的信号无关的任何不相关信息或干扰。噪声可能来源于多种因素，包括测量设备的误差、数据采集过程中的干扰、环境因素等。噪声的存在可能会对数据分析和模型建模产生负面影响，因为它可能掩盖真实的模式或关系，导致误导性的结论或预测，首先对初始数据可视化（这里选取两组）。

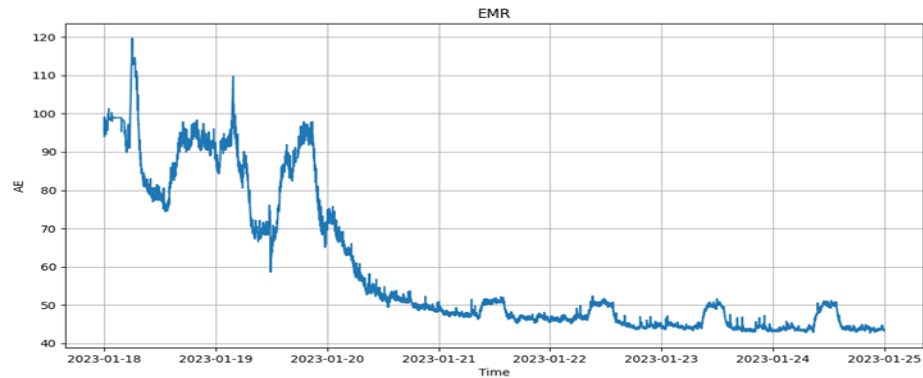


图 9：第一个时间区间的 EMR 信号可视化

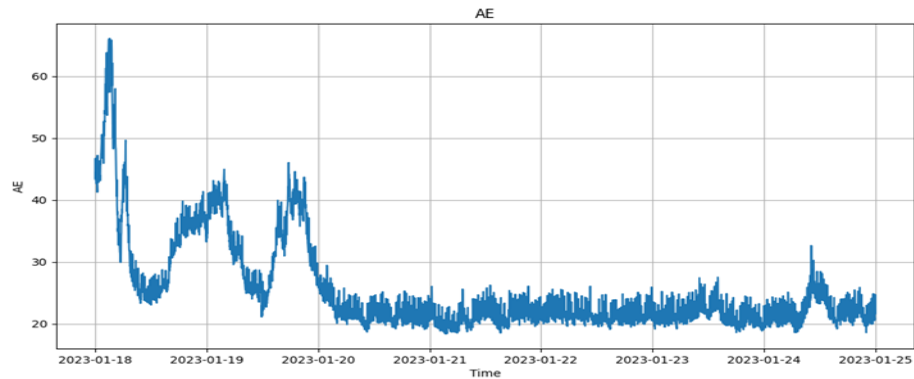


图 10：第一个时间区间的 AE 信号可视化

对此，我们采用小波去噪处理数据

- (1) 首先，将原始信号通过小波变换进行分解，得到信号在不同尺度和频率上

的小波系数。小波分解可以将信号分解成低频（近似）和高频（细节）成分；

（2）对小波系数进行阈值处理，将较小的系数设为零或者抑制它们，以滤除或减弱噪声成分。阈值处理的目标是保留真实信号的重要成分，同时去除噪声；

（3）通过对阈值处理后的小波系数进行逆变换，重构出经过去噪处理后的信号。重构的信号应当更加干净，去除了原始信号中的噪声成分。

通过小波降噪，可以同时处理时域和频域上的信息，并根据信号的特性选择不同的小波基，以更好地适应信号的结构。

在这里选取两组对小波去噪进行可视化：

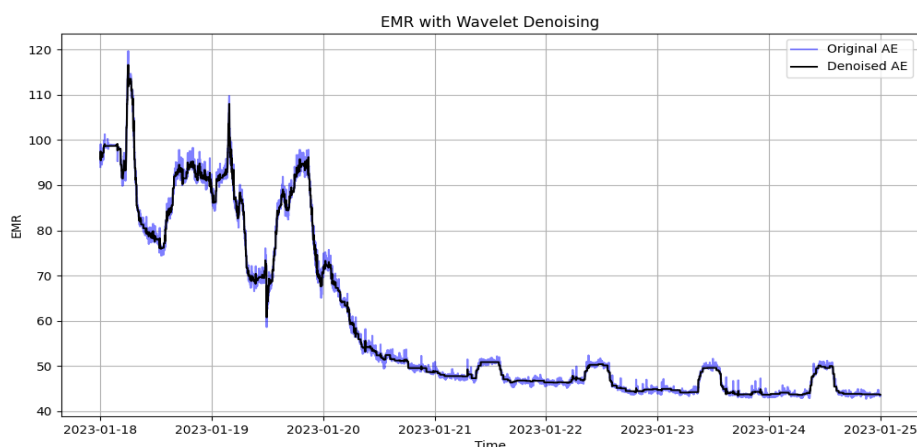


图 11：第一个时间区间的 EMR 信号小波降噪与原数据对比

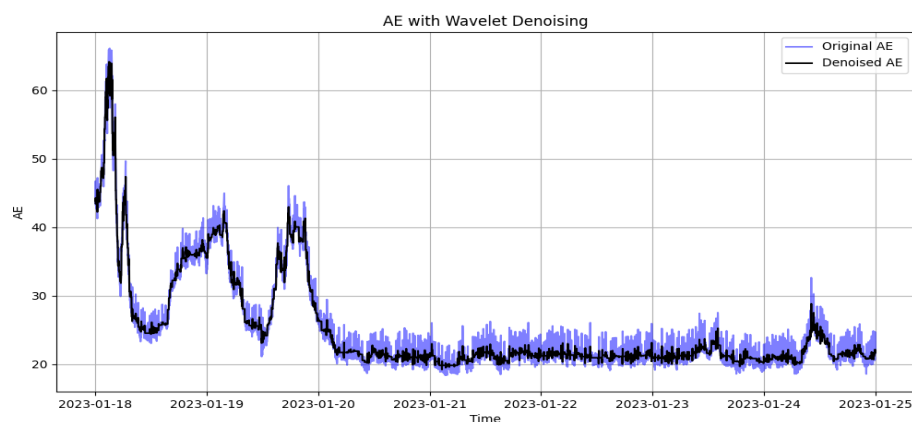


图 12：第一个时间区间的 AE 信号小波降噪与原数据对比

● Platt 缩放

Platt 缩放一种将分类模型的输出转换成概率估计的方法，具体到二分类模型，就是将模型输出的原始分数映射为 0 到 1 之间的概率值。对于我们选取的朴素贝叶斯算法，其输出的对数几率分数，使用 Platt 缩放将其转化为具体的概率值。

Platt 缩放通过对原始模型的输出分数应用一个 Sigmoid 函数（或逻辑函数）来进行转换，以便输出一个接近真实概率的值。Sigmoid 函数的公式为：

$$P(y = 1|x) = \frac{1}{1 + e^{-(A \cdot f(x) + B)}}$$

在这个公式中，是 y 目标变量， x 是特征， $f(x)$ 是模型的原始输出分数，而 A 和

B 是 Sigmoid 函数的参数，需要通过最小化对数似然损失来估计这些参数。由于参数不是由原始分类器直接学习得到的，而是通过后续过程得到的。这个后续过程通常是在原始分类模型训练完成后，通过另一步训练来完成的，我们使用 CalibratedClassifierCV 校准器，通过交叉验证来找出和的最优值。

5.3.2 朴素贝叶斯模型的建立

● 朴素贝叶斯分类器

贝叶斯定理：

朴素贝叶斯分类的基础是贝叶斯定理，该定理描述了两个条件概率之间的关系。贝叶斯定理可以表示为：

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

朴素贝叶斯分类器：

对于分类问题，我们可以用贝叶斯定理来计算给定特征集下，观测值属于某个类别的概率：

$$P(C_k|X) = \frac{P(X|C_k) \cdot P(C_k)}{P(X)}$$

由于朴素贝叶斯分类器假设所有特征之间相互独立，我们可以将条件概率简化为所有特征在类别下的概率的乘积：

$$P(X|C_k) = \prod_{i=1}^n P(x_i|C_k)$$

对于高斯朴素贝叶斯来说，假设遵循高斯分布，因此它可以用以下公式计算：

$$P(x_i|C_k) = \frac{1}{\sqrt{2\pi\sigma_{ik}^2}} \exp\left(-\frac{(x_i - \mu_{ik})^2}{2\sigma_{ik}^2}\right)$$

这里， μ_{ik} 和 σ_{ik}^2 分别是训练集中类别 C_k 下特征 x_i 的均值和方差。

当使用朴素贝叶斯模型进行预测时，我们对每个可能的类别计算新样本的后验概率，然后选择具有最高后验概率的类别作为最终的预测结果。

5.3.3 模型检验

● 学习曲线

学习曲线是一种图形工具，用于表示机器学习模型在训练过程中的学习进展。它通常随着训练样本数量的增加显示模型的性能（比如准确率、错误率或其他度量指标）是如何改变的。通过学习曲线，我们可以洞察到模型是否及在何种程度上从增加的训练数据中受益，以及模型是否面临着过拟合或欠拟合的问题。

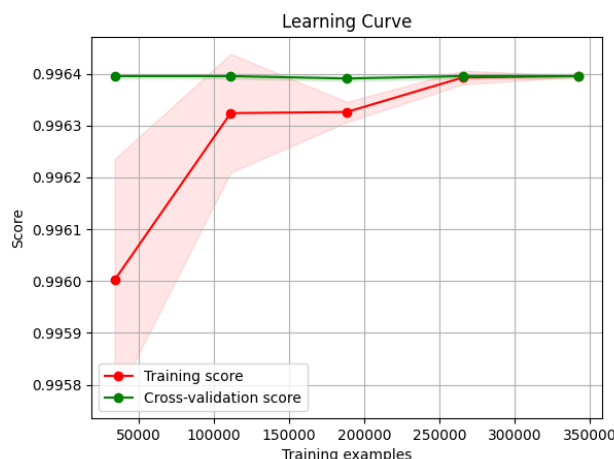


图 13: 朴素贝叶斯模型学习曲线

训练得分（红线）：最初，当数据点较少时，训练准确率非常，较少的例子使得模型更容易记忆数据。随着更多数据点的加入而保持较高准确率，表明模型有效地从额外数据中学习，没有在训练准确率上出现显著损失。

交叉验证得分（绿线）：交叉验证得分开始较低，这是意料之中的，因为模型正试图从有限的的数据中泛化。随着引入更多数据，验证集上的准确率提高，表明模型正在学习可泛化的模式。训练得分和验证得分的收敛表明，模型不仅学习良好而且能够将学到的模式泛化到未见过的数据。

由于训练得分和交叉验证得分在维持接近高准确率水平的同时收敛，且两者之间的差距不宽，学习曲线表明了一个泛化性能良好、较好适应数据的模型。

● ROC 曲线

ROC 曲线是一个效能可视化图表，用于显示分类模型在所有分类阈值下的性能。AUC 是 ROC 曲线下的面积，用一个数值来度量分类模型在区分正负样本能力方面的有效性。AUC 的值范围是 0 到 1，通常情况下，AUC 值在 0.8 到 0.9 之间被认为是一个很好的模型。

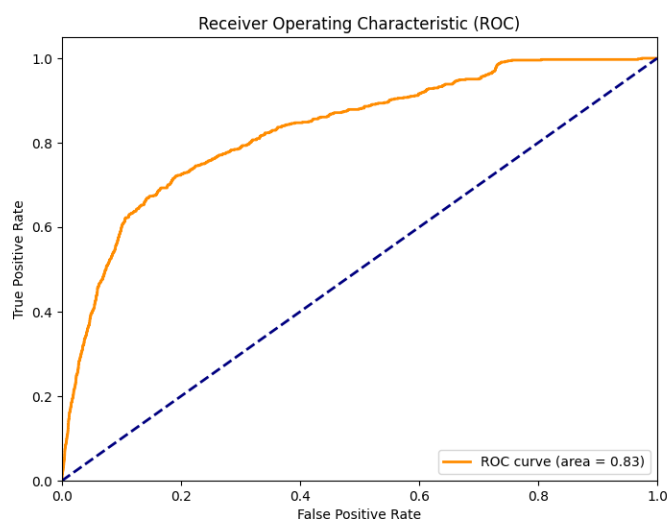


图 14: 朴素贝叶斯模型 ROC 曲线

在本模型中，AUC 为 0.83，这表明模型有相当高的区分正负样本的能力。这个值说明模型能够在大多数情况下正确地把正类和负类分开。

5.3.4 模型结论

表 5 采集数据所在时刻出现前兆特征的概率

电磁辐射数据 所在时刻	前兆特征的概率	声发射数据 所在时刻	前兆特征的概率
2023-1-24 23:58:36	0.5356	2023-1-24 23:58:36	0.0279
2023-2-11 23:59:20	0.8179	2023-2-11 23:59:20	0.1992
2023-2-26 23:59:27	0.3872	2023-2-26 23:59:27	0.8579
2023-3-10 23:58:14	0.4126	2023-3-10 23:58:14	0.8474
2023-3-30 23:58:13	0.3214	2023-3-30 23:58:13	0.6144

六、模型的优缺点与进一步讨论

6.1 模型的优点

- 性能:随机森林通常表现良好，避免了过拟合，具有较高的准确率。
- 处理不平衡数据:SMOTE 技术有效解决了类别不平衡的问题，有助于改善模型性能。
- 多样性:随机森林由多棵决策树组成，每棵树在构建时使用了不同的特征子集，这增强了模型的泛化能力。
- 简单快速:朴素贝叶斯分类器因其简单和速度快被广泛使用，尤其适用于大规模数据集和高维度问题。
- 良好的校正能力:使用 CalibratedClassifierCV 进行 Platt 缩放有助于改善概率预测，使其更可靠和解释性更强。
- 数据预处理功能:在数据预处理中使用小波降噪和滚动统计特征，有助于改进模型性能并引入更多的特征信息。

6.2 模型的缺点

- 预测速度:随机森林的预测时间比一些其他算法要长，因为它需要评估多棵决策树。
- 高维数据:当特征的数量非常多时，随机森林的效率和性能可能会降低。
- 校正方法依赖:校正模型的有效性依赖于交叉验证的适当方法和训练数据的质量。
- 对异常值敏感:由于特征的高斯模型，朴素贝叶斯可能对异常值比较敏感，这可能会影响模型的性能。

七、参考文献

- [1] Di Y, Wang E, Li Z, et al. Comprehensive early warning method of microseismic, acoustic emission, and electromagnetic radiation signals of rock burst based on deep learning [J]. International Journal of Rock Mechanics and Mining Sciences, 2023, 170: 105519.
- [2]司守奎, 孙玺菁, 《数学建模算法与应用》, 北京: 国防工业出版社, 2015. P182
- [3]孟祥宇,于春雨.深部开采冲击地压发生条件及预测和防治研究[J].科技传播,2016(17):182-183.
- [4]韩正灿,石翔元,申银朋,等.深部地质构造区冲击地压时空监测预警技术[J]. 中国矿业,2024,33(1):147-154.
- [5]梁燕华,沈奋博,谢子殿等.基于 LSTM 模型的冲击地压预测方法研究[J].中国矿业,2023, 32(5):88-95.
- [6]吴丙梅,薛生,杨超宇.基于集成学习方法的冲击地压危险性预测研究[J].矿业安全与环保,2023,50(1):54-59.

附录

第一部分：代码

代码一：时域分析指标计算：

```
%数据预处理
clc;
clear;
load DATA1.mat
%a=xlsread('附件 1 (Attachment 1).xlsx','EMR','A40948:A41498');%A 最大到 610933
%a=xlsread('550 个 AE 强度.xlsx','Sheet1','C1:C499');%A 最大到 610933

%title('627 个 C 类声波信号特征');
%xticks([0:100]);
%xticklabels({'A'});
junzhi=mean(EMR);
fangcha=var(EMR);
%计算峰度
fengzhi = kurtosis(EMR);
% 计算峰值因子
fengzhiyinzi = (max(EMR) - min(EMR)) / mean(EMR);
%计算偏度
piandu=skewness(EMR);
disp(['EMR 均值: ', num2str(junzhi)]);
disp(['EMR 方差: ', num2str(fangcha)]);
disp(['EMR 峰值: ', num2str(fengzhi)]);
disp(['EMR 峰值因子: ', num2str(fengzhiyinzi)]);
disp(['EMR 偏度: ', num2str(piandu)]);

% 使用 FFT 计算频谱
EMRpinpu = fft(EMR);
% 频谱能量
pinpunengliang = mean(abs(EMRpinpu).^2);
nengliangmidu=pinpunengliang/length(EMR);
% 主要频率成分
[max_amplitude, max_index] = max(abs(EMRpinpu));
dominant_frequency = max_index / length(EMR);

% 初始化脉冲计数
pulse_count = 0;
% 检测信号的上升沿
for i = 2:length(EMR)
    if EMR(i) > 100 && EMR(i-1) <= 100
        pulse_count = pulse_count + 1;
    end
end
```

```

        end
    end
    disp(['EMR 频谱能量: ', num2str(pinpunengliang)]);
    disp(['EMR 主要频率: ', num2str(dominant_frequency)]);
    disp(['EMR 冲击信号数目: ', num2str(pulse_count)]);
    disp(['EMR 频谱能量密度: ', num2str(nengliangmidu)]);

    junzhi2=mean(AE);
    fangcha2=var(AE);
    %计算峰度
    fengzhi2 = kurtosis(AE);
    % 计算峰值因子
    fengzhiyinzi2 = (max(AE) - min(AE)) / mean(AE);
    %计算偏度
    piandu2=skewness(AE);
    disp(['AE 均值: ', num2str(junzhi2)]);
    disp(['AE 方差: ', num2str(fangcha2)]);
    disp(['AE 峰值: ', num2str(fengzhi2)]);
    disp(['AE 峰值因子: ', num2str(fengzhiyinzi2)]);
    disp(['AE 偏度: ', num2str(piandu2)]);

    % 使用 FFT 计算频谱
    AEpinpu = fft(AE);
    % 频谱能量
    pinpunengliang2 = mean(abs(AEpinpu).^2);
    nengliangmidu2=pinpunengliang/length(AE);

    % 主要频率成分
    [max_amplitude2, max_index2] = max(abs(AEpinpu));
    dominant_frequency2 = max_index2 / length(AE);

    % 初始化脉冲计数
    pulse_count2 = 0;
    % 检测信号的上升沿
    for i = 2:length(AE)
        if AE(i) > 100 && AE(i-1) <= 100
            pulse_count2 = pulse_count2 + 1;
        end
    end
    disp(['AE 频谱能量: ', num2str(pinpunengliang2)]);
    disp(['AE 主要频率: ', num2str(dominant_frequency2)]);
    disp(['AE 冲击信号数目: ', num2str(pulse_count2)]);
    disp(['AE 频谱能量密度: ', num2str(nengliangmidu2)]);

```

代码二：频域分析指标计算：

```
%FFT 分析频谱
clc;
clear;
%附件 1 数据
%data1 = xlsread('1-2.xlsx');
%附件 2 数据
data2=xlsread('550 个 AE 强度.xlsx','Sheet1','G2:G1500');

%X=fft(data1);
Y=fft(data2);

%计算频谱
%P2 = abs(X/length(data1));
%P1 = P2(1:length(data1)/2+1);
%P1(2:end-1) = 2*P1(2:end-1);

Q2 = abs(Y/length(data2));
Q1 = Q2(1:length(data2)/2+1);
Q1(2:end-1) = 2*Q1(2:end-1);

%绘制频谱图
fs = 1000; % 采样频率
%f = fs*(0:(length(data1)/2))/length(data1);%data1 横坐标频率化
g = fs*(0:(length(data2)/2))/length(data2);%data2 横坐标频率化

%画 data1 频谱图
%plot(f, P1, '-b');
figure(1);
%画 data2 频谱图
plot(g,Q1, '-m');
title('附件 2 时间段 1AE 数据频谱图');
xlabel('频率(Hz)');
ylabel('幅度');
xlim([-10 500]);
grid on;
```

代码三：干扰信号识别：

```
clc;
clear;
close all;
load fujian2.mat;
chuang=100;
data1=AE3;
```

```

%设置标准值
stdEMR=[77.9895,8229.3701,12.9982,6.4137,3.1238,0.00019015,14305.3337];

% 设置滑动窗口
for i = 1:(length(data1)-chuang+1)
    window = data1(i:(i+chuang-1));
    % 计算窗口内数据的均值和方差
    junzhi = mean(window);
    fangcha = var(window);
    fengzhi = kurtosis(window);
    fengzhiyinzi = (max(window) - min(window)) / mean(window);
    piandu=skewness(window);
    % 使用 FFT 计算频谱
    windowpinpu = fft(window);
    % 频谱能量
    windowpinunengliang = mean(abs(windowpinpu).^2);
    windownengliangmidu = mean(abs(windowpinpu).^2)/length(window);

    % 主要频率成分
    [max_amplitude, max_index] = max(abs(windowpinpu));
    dominant_frequency = max_index / length(window);

    % 初始化脉冲计数
    pulse_count = 0;
    % 检测信号的上升沿
    for j = 2:length(window)
        if window(j) > 100 && window(j-1) <= 100
            pulse_count = pulse_count + 1;
        end
    end
    %指标: 均值, 方差, 峰值, 峰值因子, 偏度, 能量密度, 主要频率

    result(i,:)= [junzhi,fangcha,fengzhi,fengzhiyinzi,piandu,dominant_frequency,windowne
ngliangmidu];
end
disp('EMR 时间段 1 各项指标: ');
disp(result);

% 判断窗口内数据是否满足指标
% 设置误差比例
error_rate=0.1;
givenMatrix = stdEMR;
dataMatrix=result;
% 初始化一个空数组, 用于存储符合条件的行的位置

```

```

matchedRows = [];
% 遍历 351*5 矩阵的每一行
for i = 1:size(dataMatrix, 1)
% 检查当前行的每一列是否在给定矩阵的对应列上下 0.1 区间内
if any(dataMatrix(i, :) < givenMatrix*(1+error_rate) & dataMatrix(i, :) >
givenMatrix*(1-error_rate))
% 如果满足条件, 则将当前行的位置存储在 matchedRows 数组中
matchedRows = [matchedRows, i];
end
end
% 输出符合条件的行的位置
disp('EMR 时间段 1 干扰信号时间起点所在行位置: ');
disp((matchedRows-1)*chuang+1);

```

第二部分：代码

代码一：问题二对前兆特征数据的特征提取（以 EMR 信号为例）：

```

import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_excel('Attachment 1.xlsx', sheet_name='EMR')
# 提取类别 B 的电磁辐射数据
data_b_emr = data[(data['class'] == 'B') & (data['EMR'].notna())]
# 绘制电磁辐射数据的时间序列图
plt.figure(figsize=(14, 7))
plt.plot(data_b_emr['time'], data_b_emr['EMR'], marker='o', linestyle='-', markersize=2, label='class B', color='red')
plt.title('Electromagnetic emission trends for precursory features')
plt.xlabel('time')
plt.ylabel('EMR')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# 计算变化速率
data_b_emr['change_rate'] = data_b_emr['EMR'].diff()

# 计算滑动平均和标准偏差（滑动窗口大小设为 30，约 15 分钟数据）
window_size = 30
data_b_emr['rolling_mean'] = data_b_emr['EMR'].rolling(window=window_size).mean()
data_b_emr['rolling_std'] = data_b_emr['EMR'].rolling(window=window_size).std()

# 重新绘制电磁辐射及其趋势特征图
plt.figure(figsize=(14, 10))

```

```

# 原始电磁辐射数据
plt.subplot(3, 1, 1)
plt.plot(data_b_emr['time'], data_b_emr['EMR'], label='Primary Electromagnetic emission')
plt.title('Electromagnetic emission and its trend characteristics')
plt.ylabel('EMR')
plt.legend()

# 变化速率
plt.subplot(3, 1, 2)
plt.plot(data_b_emr['time'], data_b_emr['change_rate'], label='Rate of change', color='orange')
plt.ylabel('Rate of change')
plt.legend()

# 滑动平均与标准偏差
plt.subplot(3, 1, 3)
plt.plot(data_b_emr['time'], data_b_emr['rolling_mean'], label='Moving average', color='green')
plt.plot(data_b_emr['time'], data_b_emr['rolling_std'], label='Sliding standard deviation', color='red')
plt.xlabel('time')
plt.ylabel('Moving average/standard deviation')
plt.legend()

plt.tight_layout()
plt.show()
#关注异常峰值、剧烈波动的变化率（一阶差分）、剧烈波动的滑动均值/标准差等参量
#引入频率域

# 计算变化速率
data_b_emr['change_rate'] = data_b_emr['EMR'].diff()

# 计算滑动平均和标准偏差（滑动窗口大小设为 30，约 15 分钟数据）
window_size = 30
data_b_emr['rolling_mean'] = data_b_emr['EMR'].rolling(window=window_size).mean()
data_b_emr['rolling_std'] = data_b_emr['EMR'].rolling(window=window_size).std()

# 重新绘制电磁辐射及其趋势特征图
plt.figure(figsize=(14, 10))

# 原始电磁辐射数据
plt.subplot(3, 1, 1)
plt.plot(data_b_emr['time'], data_b_emr['EMR'], label='Primary Electromagnetic emission')
plt.title('Electromagnetic emission and its trend characteristics')
plt.ylabel('EMR')
plt.legend()

```

```

# 变化速率
plt.subplot(3, 1, 2)
plt.plot(data_b_emr['time'], data_b_emr['change_rate'], label='Rate of change', color='orange')
plt.ylabel('Rate of change')
plt.legend()

# 滑动平均与标准偏差
plt.subplot(3, 1, 3)
plt.plot(data_b_emr['time'], data_b_emr['rolling_mean'], label='Moving average', color='green')
plt.plot(data_b_emr['time'], data_b_emr['rolling_std'], label='Sliding standard deviation', color='red')
plt.xlabel('time')
plt.ylabel('Moving average/standard deviation')
plt.legend()

plt.tight_layout()
plt.show()
#关注异常峰值、剧烈波动的变化率（一阶差分）、剧烈波动的滑动均值/标准差等参量
#引入频率域

#最终的特征向量取电磁辐射(EMR)、fft_mean（傅里叶变换的平均值）、fft_std（傅里叶变换的标准差）、
fft_max（傅里叶变换的最大值）
#d1（一阶差分）、rolling_mean（滑动均值）、rolling_var（滑动方差）、rolling_std（滑动标准差）
train_data = pd.read_excel('Attachment 1.xlsx', sheet_name='EMR')
pro_data = pre(train_data)
pro_data

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

# 绘制傅里叶特征
plt.figure(figsize=(12, 3))
plt.subplot(1, 3, 1)
sns.distplot(pro_data['fft_mean'], kde=True, color='skyblue')
plt.title('FFT Mean Distribution')

plt.subplot(1, 3, 2)
sns.distplot(pro_data['fft_std'], kde=True, color='salmon')
plt.title('FFT Standard Deviation Distribution')

plt.subplot(1, 3, 3)
sns.distplot(pro_data['fft_max'], kde=True, color='green')
plt.title('FFT Maximum Value Distribution')

```

```
plt.tight_layout()
plt.show()
#关注傅里叶特征的频率聚集
```

代码二：问题二随机森林分类模型的构建、训练与评价（以 AE 信号为例）：

```
#添加前兆信号标签
```

```
pro_data['前兆信号'] = 0
pro_data.loc[pro_data['类别 (class)'] == 'B', '前兆信号'] = 1
pro_data
```

```
#数据集调优（过采样）
```

```
from imblearn.over_sampling import SMOTE
import pandas as pd
```

```
X = pro_data.drop(['时间 (time)', '类别 (class)', '前兆信号'], axis=1)
y = pro_data['前兆信号']
```

```
# 显示数据集类别分布
print("Class distribution before SMOTE:")
print(y.value_counts())
```

```
# 使用 SMOTE 算法进行过采样
smote = SMOTE(random_state=60063)
X_resampled, y_resampled = smote.fit_resample(X, y)
```

```
# 创建过采样后的 DataFrame
pro_data_resampled = pd.DataFrame(X_resampled, columns=X.columns)
pro_data_resampled['前兆信号'] = y_resampled
```

```
# 显示过采样后的类别分布
print("\nClass distribution after SMOTE:")
print(pro_data_resampled['前兆信号'].value_counts())
```

```
#随机森林拟合
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
X = pro_data_resampled.drop(['前兆信号'], axis=1)
y = pro_data_resampled['前兆信号']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=60063)
model = RandomForestClassifier(n_estimators=100, random_state=60063)
```

```

model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
conf_mat = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", conf_mat)

#混淆矩阵展示
plt.figure(figsize=(8, 6))
sns.heatmap(conf_mat, annot=True, fmt="d", cmap="Blues")
plt.xlabel("Predicted labels")
plt.ylabel("True labels")
plt.title("Confusion Matrix")
plt.show()

from sklearn.metrics import roc_curve, roc_auc_score
# 计算 ROC 曲线数据
y_pred_proba = model.predict_proba(X_test)[:, 1]
fpr, tpr, thresholds = roc_curve(y_test, y_pred_proba)
roc_auc = roc_auc_score(y_test, y_pred_proba)

# 绘制 ROC 曲线
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='blue', label='ROC curve (AUC = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], linestyle='--', color='black')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('AE Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.grid(True)
plt.show()

代码三：问题二分类模型在题设数据的拟合与预测（以 AE 信号为例）：
#2021 年 11 月 1 日-2022 年 1 月 15 日声发射信号求解
unknown_data = pd.read_excel('AE.xlsx')
goal_data = pre(unknown_data)
X_goal = goal_data.drop('时间 (time)', axis=1)
X_goal

#模型预测
y_pred = model.predict(X_goal)

```

```

goal_data['前兆信号'] = y_pred
goal_data

#查询函数
def find(goal_data):
    goal_data['start'] = (goal_data['前兆信号'] == 1) & (goal_data['前兆信号'].shift(1) != 1)
    goal_data['end'] = (goal_data['前兆信号'] == 1) & (goal_data['前兆信号'].shift(-1) != 1)

    breaktime = []
    start_time = None
    for _, row in goal_data.iterrows():
        if row['start']:
            start_time = row['时间 (time)']
        if row['end'] and start_time is not None:
            end_time = row['时间 (time)']
            breaktime.append([start_time, end_time])
            start_time = None

    breaktime_data = pd.DataFrame(breaktime)
    breaktime_data.to_csv('answer3.csv', index = False)#文件写入
    print("Success")
    return breaktime_data

in_data = find(goal_data)

```

第三部分：代码

```

import pandas as pd
import numpy as np
import os
import pywt

from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.calibration import CalibratedClassifierCV

# 读取数据
print("Current working directory:", os.getcwd())

full_file_path = 'C:\\Users\\wei jingxuan\\Desktop\\Attachment 1.xlsx'
data = pd.read_excel(full_file_path, sheet_name='AE')

# 小波降噪函数
def wavelet_denoising(signal, wavelet='db1', level=1):

```

```

coeff = pywt.wavedec(signal, wavelet, mode="per", level=level)
sigma = (1 / 0.6745) * mad(coeff[-level])
uthresh = sigma * np.sqrt(2 * np.log(len(signal)))
coeff[1:] = (pywt.threshold(i, value=uthresh, mode='soft') for i in coeff[1:])
reconstructed_signal = pywt.waverec(coeff, wavelet, mode='per')
reconstructed_signal = reconstructed_signal[:len(signal)] # 裁剪到原信号长度
if len(reconstructed_signal) < len(signal):
    # 如果重构信号小于原始信号长度，进行填充
    reconstructed_signal = np.pad(reconstructed_signal, (0, len(signal) - len(reconstructed_signal)), 'edge')
return reconstructed_signal

# 计算中位绝对偏差
def mad(data, axis=None):
    return np.median(np.absolute(data - np.median(data, axis)), axis)

# 预处理函数
def pre(data):
    def cal_fft_features(signal):
        if isinstance(signal, pd.Series):
            signal = signal.values
        if isinstance(signal, (list, tuple)):
            signal = np.array(signal)
        if isinstance(signal, (float, int)):
            signal = np.array([signal])
        if signal.ndim > 1:
            signal = signal.squeeze()

        fft_values = np.fft.fft(signal)
        fft_abs = np.abs(fft_values)
        return np.mean(fft_abs), np.std(fft_abs), np.max(fft_abs)

    # 在预处理流程中添加降噪步骤
    signal_denoised = wavelet_denoising(data['声波强度 (AE)'].values)
    data['声波强度 (AE)_denoised'] = signal_denoised

    # 使用降噪后的信号进行后面的计算
    fft_features = data['声波强度 (AE)_denoised'].apply(cal_fft_features)
    data['fft_mean'], data['fft_std'], data['fft_max'] = zip(*fft_features)

    data['d1'] = data['声波强度 (AE)_denoised'].diff(1).fillna(0)
    data['rolling_mean'] = data['声波强度 (AE)_denoised'].rolling(window=10).mean().bfill()
    data['rolling_std'] = data['声波强度 (AE)_denoised'].rolling(window=10).std().bfill()
    return data

```

```

# 预处理数据
pro_data = pre(data)
# 定义特征和标签
features_list = ['声波强度 (AE)', 'fft_mean', 'fft_std', 'fft_max', 'd1', 'rolling_mean', 'rolling_std']

# 假设数据集中的 '类别 (class)' 列包含用于预测的目标类别
# B 类别代表预兆信号
pro_data['前兆信号'] = 0
pro_data.loc[pro_data['类别 (class)'] == 'B', '前兆信号'] = 1

X = pro_data[features_list]
y = pro_data['前兆信号']

# 划分训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# 初始化朴素贝叶斯模型并拟合训练集
model = GaussianNB()
model.fit(X_train, y_train)

# 使用 CalibratedClassifierCV 对模型进行 Platt 缩放
calibrated_model = CalibratedClassifierCV(model, method='sigmoid', cv=5) # 使用交叉验证来训练校正器
calibrated_model.fit(X_train, y_train)

# 使用校正的模型对测试集进行预测
y_pred = calibrated_model.predict(X_test)
y_pred_proba = calibrated_model.predict_proba(X_test)

# 输出校正模型的预测结果准确率
accuracy_calibrated = accuracy_score(y_test, y_pred)
print(f'Calibrated Accuracy: {accuracy_calibrated}')

# 接下来对附件 3 中的数据进行预测
predict_data = pd.read_excel('path') # 加载附件 3 数据
pro_predict_data = pre(predict_data) # 对附件 3 数据应用相同的预处理函数
# 使用校正后的模型进行概率预测
predict_proba_calibrated = calibrated_model.predict_proba(pro_predict_data[features_list])

# 输出校正后的预测概率
# 假设'时间 (time)'是附件 3 数据集中时间戳的列名
for i in range(len(predict_proba_calibrated)):
    print(f'Time:      {pro_predict_data.iloc[i]["time"]}      -      Precursory      Signal      Probability:
{predict_proba_calibrated[i][1]}')

```