

LSM6DSOX: 机器学习内核

引言

本文档旨在提供 **LSM6DSOX** 中可用的机器学习内核功能信息。机器学习处理能力允许将一些算法从应用处理器转移到 MEMS 传感器，从而持续降低功耗。

通过决策树逻辑获得机器学习处理能力。决策树是由一系列可配置节点组成的数学工具。每个节点均采用“如果-则-否则”条件，并根据阈值评估输入信号（由传感器数据计算出的统计参数表示）。

可通过配置 **LSM6DSOX** 来同时独立运行多达 8 个决策树。决策树存储在设备中，并在专用输出寄存器中生成结果。

可通过应用处理器随时读取决策树的结果。此外，可针对决策树结果的每次更改产生中断。

图 1. 监督式机器学习方法



1 中的机器学习内核 LSM6DSOX

机器学习内核（以及有限状态机）是 LSM6DSOX 中的主要嵌入式功能之一。它由能够在传感器本身中实现算法的一组可配置的参数和决策树组成。

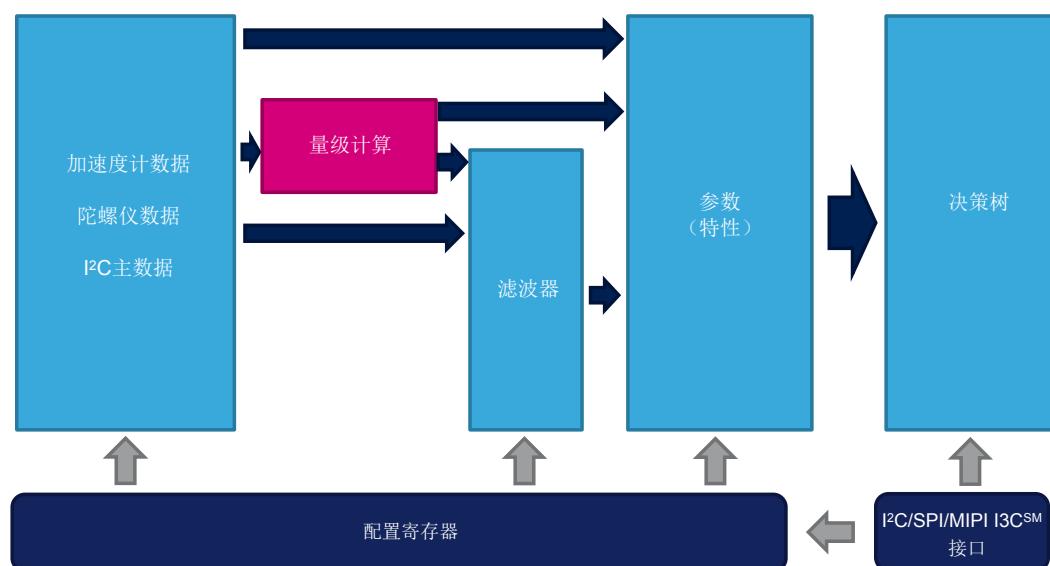
适用于机器学习内核的算法为可通过归纳法实现的算法，这涉及从观察结果搜索模式。采用该方法的一些算法示例包括：活动识别、健身活动识别、运动强度检测、振动强度检测、携带位置识别、情景感知、假阳性拒绝等...

机器学习内核的理念为使用加速度计、陀螺仪和外部传感器数据（可通过 I²C 主接口读取）来计算一组可由用户在定义的时间窗口中选择的统计参数（例如均值、方差、能量、峰值、过零信号等...）。除了传感器输入数据以外，还可以通过应用设备中可用的一些可配置滤波器来定义一些新输入。

机器学习内核参数被称为“功能”，可用作可配置决策树的输入，该决策树可存储在设备中。

可存储在 LSM6DSOX 中的决策树为由一系列节点组成的二叉树。在每个节点中，根据阈值评估统计参数（特征），以建立下一个节点中的演化。当达到叶时（树中的最后一个节点之一），决策树产生结果，可通过专用设备寄存器来读取该结果。

图 2. 中的机器学习内核 LSM6DSOX



可将机器学习内核输出数据配置为 12.5 到 104 Hz 之间的四个可用速率之一。嵌入式功能寄存器 EMB_FUNC_ODR_CFG_C (60h) 中的 MLC_ODR 位允许选择四个可用速率之一，如下表所示。

表 1. 机器学习内核输出数据速率

EMB_FUNC_ODR_CFG_C (60h) 中的 MLC_ODR 位	机器学习内核输出数据速率
00	12.5 Hz
01	26 Hz (默认值)
10	52 Hz
11	104 Hz

为实现 LSM6DSOX 的机器学习处理能力，必须使用“监督式学习”方法，该方法包括：

- 确定一些要识别的类别；
- 为每个类别收集多个数据日志；

- 通过收集的日志执行一些数据分析，以学习通用规则，该规则允许将输入（数据日志）映射到输出（要识别的类别）。

例如，在活动识别算法中，要识别的类别可能为：静止、行走、慢跑、骑自行车、驾驶等... 必须为每个类别获取多个数据日志，例如，多人执行相同的活动。

分析收集的数据日志的目的：

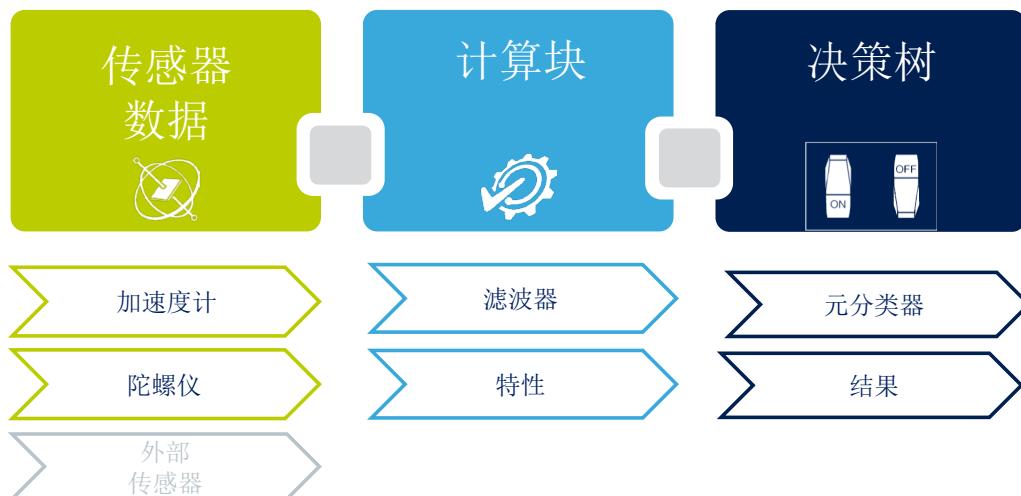
- 定义用于对不同的类别进行正确分类的功能；
- 定义将应用于输入数据的滤波器，以使用所选特征提高性能；
- 生成能够识别不同类别之一的专用决策树（将输入映射到输出）。

一旦定义了决策树，即可通过意法半导体提供的软件工具（如第 2 节 机器学习内核工具中所述）生成设备配置。决策树将在设备上运行，以最大限度地降低功耗。

要进一步详细了解 LSM6DSOX 中的机器学习内核功能，可以将其视为三个主要模块(图 3)：

- 传感器数据
- 计算块
- 决策树

图 3. 机器学习内核块



第一个模块被称为“传感器数据”，该数据包括内置于设备的加速度计和陀螺仪的数据，或可能通过 I²C 主接口（传感器集线器）连接到 LSM6DSOX 的外部传感器的数据。

第一个块中定义的机器学习内核输入用于第二个块，即“计算块”，可在其中应用滤波器和特征。特征为定义的时间窗口中的输入数据（或滤波后的数据）计算出的统计参数，用户可选择这些参数。

在计算块中计算出的特征将用作机器学习内核的第三个块的输入。该块被称为“决策树”，这包括二叉树，该二叉树用于评估通过输入数据计算出的统计参数。在二叉树中，通过将统计参数与特定阈值进行比较来生成结果（在上述活动识别示例中，结果为：静止、行走、慢跑、骑自行车等...）。也可以通过被称为“元分类器”的可选滤波器过滤决策树结果。机器学习内核结果将为决策树结果，其中包括可选的元分类器。

机器学习内核存储器以“动态”或“模块化”方式组织，以便最大限度地增加可在设备（滤波器、功能等）中配置的计算块的数量。已设计用于生成 LSM6DSOX 的配置的专用工具，以自动管理存储器使用。该工具在 Unico GUI 中可用，稍后将在第 2 节 机器学习内核工具中介绍。

下节详细说明了图 3 中所述的 LSM6DSOX 中的机器学习内核的三个主要模块。

1.1 输入

LSM6DSOX 用作组合（加速度计 + 陀螺仪）传感器，可生成加速度和角速率输出数据。加速度和角速率的 3 轴数据可用作机器学习内核的输入。图 4 和图 5 显示了加速度计和陀螺仪数字链中的机器学习内核块输入。对于 LSM6DSOX 中可用的全部四种连接模式，机器学习内核（MLC）块在两个数字链中的位置相同。

图 4. MLC 输入（加速度计）

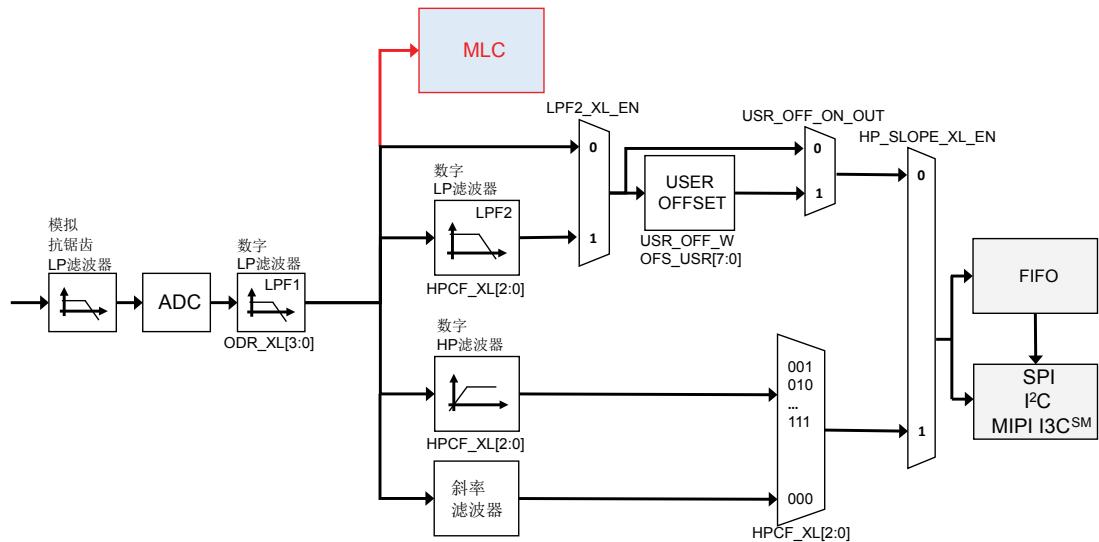
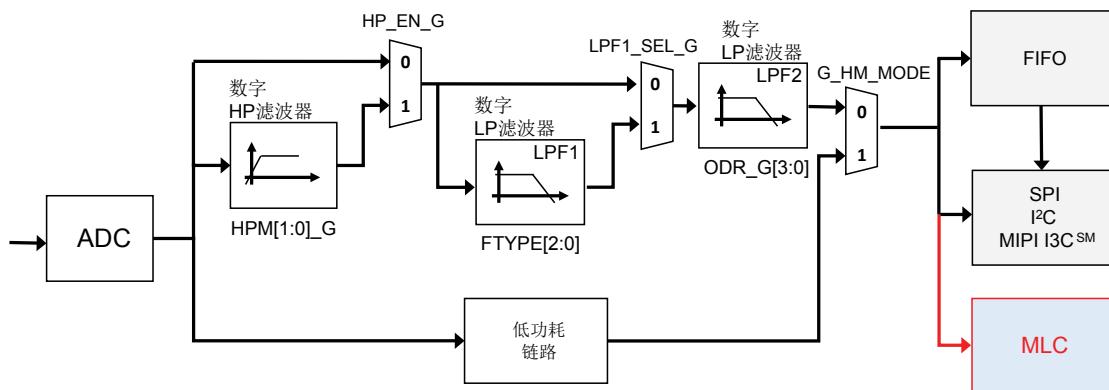


图 5. MLC 输入（陀螺仪）



输入数据的速率必须等于或高于嵌入式功能寄存器 EMB_FUNC_ODR_CFG_C (60h)配置的机器学习内核数据速率，如表 1 中所述。

示例：在以 26 Hz 运行的活动识别算法中，必须选择 26 Hz 的机器学习内核 ODR，而传感器 ODR 必须等于或高于 26 Hz。

机器学习内核使用采用单位：

- 加速度计数据的单位为 [g]
- 陀螺仪数据的单位为 [rad/sec]
- 外部传感器数据的单位为 [Gauss]，压力传感器的单位为 [Bar]

由于可通过传感器集线器功能（模式 2）将外部传感器（如磁力计）连接至 LSM6DSOX，因此来自外部传感器的数据也可以用作机器学习处理的输入。

使用外部传感器时，必须通过寄存器 MLC_MAG_SENSITIVITY_L (E8h)和 MLC_MAG_SENSITIVITY_H (E9h)设置外部传感器的灵敏度。

例如：对于诸如 LIS2MDL 的磁力计，灵敏度为 1.5 mG/LSB，将其转换为高斯时，灵敏度变为 0.0015 G/LSB，将其转换为 HFP (LSM6DSOX 灵敏度寄存器的半精度浮点值) 时，灵敏度变为 1624h。

灵敏度[mG/LSB]	灵敏度[G/LSB]	灵敏度 HFP
1.5 mG/LSB	0.0015 G/LSB	1624h

注：半精度浮点格式表示为：

SEEEEEFFFFFFF (S: 1 个符号位; E: 5 个指数位; F: 10 个分数位)。

以下程序允许更改外部磁力计数据的转换系数：

1. 将 80h 写入寄存器 01h //使能对嵌入功能寄存器的访问
2. 将 40h 写入寄存器 17h // PAGE_RW (17h) = '40h': 使能写操作
3. 将 11h 写入寄存器 02h // PAGE_SEL (02h) = '11h'
4. 将 E8h 写入寄存器 08h // PAGE_ADDRESS (08h) = 'E8h'
5. 将[LSB]转换系数 (LIS2MDL 示例, 24h) 写入寄存器 09h
6. 将 11h 写入寄存器 02h // PAGE_SEL (02h) = '11h'
7. 将 E9h 写入寄存器 08h // PAGE_ADDRESS (08h) = 'E9h'
8. 将[MSB]转换系数 (LIS2MDL 示例, 16h) 写入寄存器 09h
9. 将 00h 写入寄存器 17h // PAGE_RW (17h) = '00h': 禁用读/写操作
10. 将 00h 写入寄存器 01h // 禁用对嵌入功能寄存器的访问

机器学习内核工具（如第 2 节 机器学习内核工具中所述）生成的配置中包含上述更改外部传感器灵敏度的程序示例，因此用户只需在 GUI 中设置灵敏度值，软件就会将值转换为寄存器设置。

汇总机器学习内核输入：

- 加速度计数据转换系数由设备自动处理；
- 陀螺仪数据转换系数由设备自动处理；
- 设备不自动处理外部传感器数据转换系数。用户必须设置转换系数，以使机器学习内核能够使用正确的测量单位。

适用于所有传感器数据（加速度计、陀螺仪和外部传感器）的附加输入都是范数。机器学习内核（位于 LSM6DSOX 中）通过 3 轴数据在内部计算范数和范数平方。这两个附加信号可用作机器学习处理的输入。

使用以下公式计算输入数据的范数和范数平方：

$$V = \sqrt{x^2 + y^2 + z^2}$$

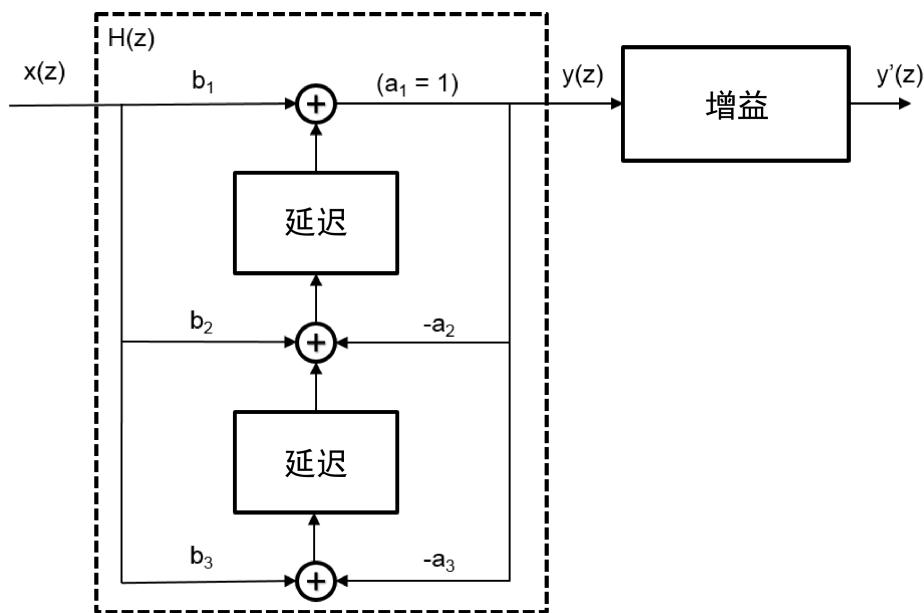
$$V^2 = x^2 + y^2 + z^2$$

范数与范数平方数据可用于决策树，以确保为用户提供高水平的程序定制。

1.2 滤波器

可通过机器学习内核逻辑中可用的各种滤波器来过滤上一节中的输入数据。机器学习内核滤波的基本元素为二阶 IIR 滤波器，如下图所示。

图 6. 滤波器基本元素



通用 IIR 2 阶滤波器的转换功能如下：

$$H(z) = \frac{b_1 + b_2 z^{-1} + b_3 z^{-2}}{1 + a_2 z^{-1} + a_3 z^{-2}}$$

通过图 6，可将输出定义为：

$$\begin{aligned} y(z) &= H(z) \cdot x(z) \\ y'(z) &= y(z) \cdot Gain \end{aligned}$$

为优化内存使用，机器学习内核具有针对不同类型的滤波器（高通、带通、IIR1、IIR2）的默认系数。在选择滤波器类型后，机器学习内核工具将通过询问所需的滤波器系数来帮助配置滤波器。下表显示了取决于所选滤波器类型的系数默认值和可配置值。通过设置不同的系数，可针对特定应用调整滤波器。

表 2. 滤波系数

滤波器类型/系数	b ₁	b ₂	b ₃	a ₂	a ₃	增益
高通滤波器	0.5	-0.5	0	0	0	1
带通滤波器	1	0	-1	可配置	可配置	可配置
IIR1 滤波器	可配置	可配置	0	可配置	0	1
IIR2 滤波器	可配置	可配置	可配置	可配置	可配置	1

滤波系数值表示为半精度浮点格式：S EEEEEEFFFFFFFFF (S: 1 个符号位；E: 5 个指数位；F: 10 个分数位)。

1.2.1 滤波系数

可使用不同的工具（包括 Matlab、Octave 和 Python）计算 IIR 滤波器系数。例如，在 Matlab 中，以下函数可用于生成低通滤波器的系数：

```
[b, a] = butter( N, f_cut / (ODR/2), 'low' )
```

其中：

- N 为 IIR 滤波器的阶数 (IIR1 为 1, IIR2 为 2)
- f_cut 为滤波器的截止频率[Hz]

- ODR 为机器学习内核数据速率[Hz]
- “low”（或“high”）为要实施的滤波器类型（低通或高通）

注意：可配置一个高通滤波器，其截止频率为带宽（ $ODR/4$ ）的一半，而无需插入系数。机器学习内核具有一些针对该配置的预定义系数。

以下函数允许通过 Matlab 生成带通滤波器系数：

```
[b,a] = butter(1,[f1 f2]/(ODR/2), 'bandpass')
```

注意：由于带通滤波器仅可配置 a_2 、 a_3 和 Gain，因此应通过设置 $Gain = b(1)$ 来标准化 b 向量。

示例：

$b = [0.2929 \ 0 \ -0.2929]; \ a = [1.0 \ -0.5858 \ 0.4142];$

可写成 $b = [1 \ 0 \ -1]$ 和 $Gain = 0.2929$ 。

因此，带通滤波器的系数为：

$a_2 = -0.5858; \ a_3 = 0.4142; \ Gain = 0.2929$ 。

下表显示了一些滤波器系数的示例（大多数示例考虑了 26 Hz 的 ODR）。

表 3. 滤波器系数示例

滤波器类型/系数	b_1	b_2	b_3	a_2	a_3	增益
高通 IIR1, $f_{cut} = 1$ Hz, $ODR = 26$ Hz	0.891725	-0.891725	-	-0.783450	-	1
高通 IIR1, $f_{cut} = 2$ Hz, $ODR = 26$ Hz	0.802261	-0.802261	-	-0.604521	-	1
高通 IIR1, $f_{cut} = 5$ Hz, $ODR = 26$ Hz	0.591628	-0.591628	-	-0.183257	-	1
高通 IIR1, $f_{cut} = 10$ Hz, $ODR = 26$ Hz	0.274968	-0.274968	-	0.450063	-	1
高通 IIR2, $f_{cut} = 1$ Hz, $ODR = 26$ Hz	0.8428435	-1.685687	0.8428435	-1.6608344	0.710540	1
高通 IIR2, $f_{cut} = 2$ Hz, $ODR = 26$ Hz	0.709560	-1.419120	0.709560	-1.332907	0.505334	1
高通 IIR2, $f_{cut} = 5$ Hz, $ODR = 26$ Hz	0.4077295	-0.815459	0.407730	-0.426937	0.203981	1
高通 IIR2, $f_{cut} = 10$ Hz, $ODR = 26$ Hz	0.085605	-0.171209	0.085605	1.019146	0.361564	1
低通 IIR1, $f_{cut} = 1$ Hz, $ODR = 26$ Hz	0.108275	0.108275	-	-0.783450	-	1
低通 IIR1, $f_{cut} = 2$ Hz, $ODR = 26$ Hz	0.197739	0.197739	-	-0.604521	-	1
低通 IIR1, $f_{cut} = 5$ Hz, $ODR = 26$ Hz	0.408372	0.408372	-	-0.183257	-	1
低通 IIR1, $f_{cut} = 10$ Hz, $ODR = 26$ Hz	0.725032	0.725032	-	0.450063	-	1
低通 IIR2, $f_{cut} = 1$ Hz, $ODR = 26$ Hz	0.012426	0.024853	0.012426	-1.660834	0.710540	1
低通 IIR2, $f_{cut} = 2$ Hz,	0.043107	0.086213	0.043107	-1.332907	0.505333	1

滤波器类型/系数	b ₁	b ₂	b ₃	a ₂	a ₃	增益
ODR = 26 Hz						
低通 IIR2, f _{cut} = 5 Hz, ODR = 26 Hz	0.194261	0.388522	0.194261	-0.426937	0.203981	1
低通 IIR2, f _{cut} = 10 Hz, ODR = 26 Hz	0.595178	1.190355	0.595178	1.019146	0.361564	1
带通 IIR2, f ₁ = 1.5 Hz, f ₂ = 5 Hz, ODR = 26 Hz	0.310375	0	-0.310375	-1.069500	0.379250	1
带通 IIR2, f ₁ = 0.2 Hz, f ₂ = 1 Hz, ODR = 100 Hz	0.0236	0	-0.0236	-1.9521	0.9528	1

1.3 特征

这些特征为通过机器学习内核输入计算出的统计参数。可用作特征计算的机器学习内核输入为：

- 传感器输入，包括
 - X、Y、Z 轴的传感器数据（如 Acc_X、Acc_Y、Acc_Z、Gyro_X、Gyro_Y、Gyro_Z）；
 - 外部传感器数据（如 ExtSens_X、ExtSens_Y、ExtSens_Z）；
 - 传感器/外部传感器数据的范数和范数平方信号（Acc_V、Acc_V2、Gyro_V、Gyro_V2、ExtSens_V、Ext_Sens_V2）；
- 滤波数据（如 Acc_Z 上的高通、Acc_V2 上的带通等）

所有特征均在定义的时间窗口内计算，这也被称为“窗口长度”，因为它表示样本数。窗口大小必须由用户确定，这对于机器学习处理非常重要，因为决策树中的所有统计参数均将在该时间窗口中评估。这不是移动窗口，对于每个 WL 样本，只计算一次特征（其中，WL 为窗口大小）。

窗口长度可能为 1 到 255 个样本的值。窗口长度值的选择取决于传感器数据速率（ODR），这将引入用于产生机器学习内核结果的等待时间，同时也取决于特定应用程序或算法。例如，在活动识别算法中，可决定每隔 2 秒或 3 秒计算一次特征，这意味着，考虑到传感器以 26 Hz 运行，窗口长度应分别约为 50 或 75 个样本。

评估机器学习内核中的某些特征需要一些附加参数（如附加阈值）。下表显示了机器学习内核中的所有可用特征，包括附加参数。

表 4. 特征

特征	附加参数
均值	-
方差	-
能量	-
峰到峰	-
过零信号	阈值
正过零信号	阈值
负过零信号	阈值
峰值探测器	阈值
正峰值探测器	阈值
负峰值探测器	阈值
最小值	-
最大值	-

1.3.1

均值

“均值”特征通过以下公式计算定义的时间窗口(WL)中的所选输入(I)的均值：

$$Mean = \frac{1}{WL} \sum_{k=0}^{WL-1} I_k$$

1.3.2

方差

“方差”特征通过以下公式计算定义的时间窗口(WL)中的所选输入(I)的方差：

$$Variance = \left(\frac{\sum_{k=0}^{WL-1} I_k^2}{WL} \right) - \left(\frac{\sum_{k=0}^{WL-1} I_k}{WL} \right)^2$$

1.3.3

能量

“能量”特征通过以下公式计算定义的时间窗口(WL)中的所选输入(I)的能量：

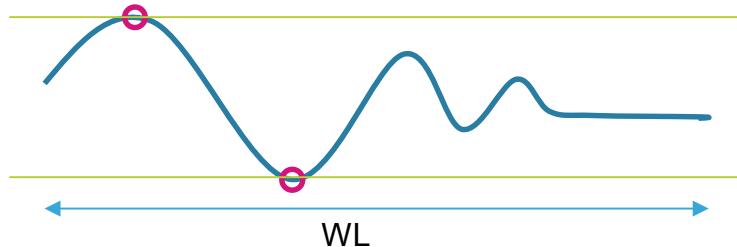
$$Energy = \sum_{k=0}^{WL-1} I_k^2$$

1.3.4

峰到峰

“峰到峰”特征可计算定义的时间窗口中的所选输入的最大峰峰值。

图 7. 峰到峰

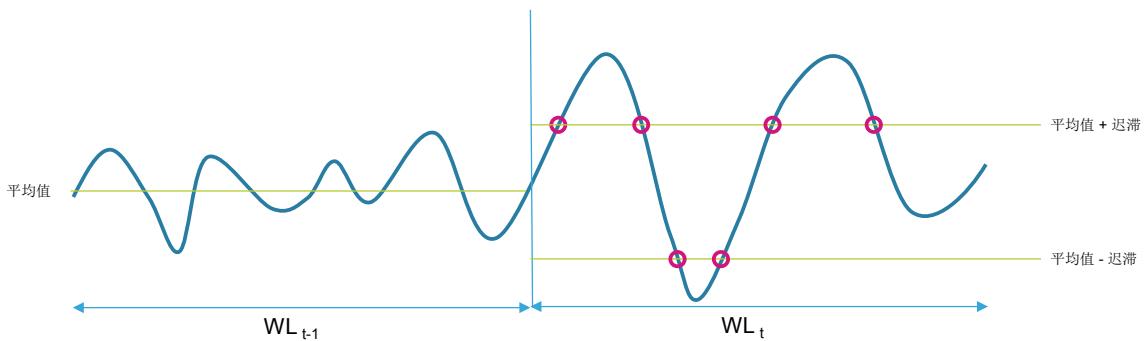


1.3.5

过零信号

“过零信号”特征可计算所选输入跨过某个阈值的次数。该内部阈值定义为在前一个窗口中计算的均值（“均值”特征）与用户定义的迟滞之和。

图 8. 过零信号

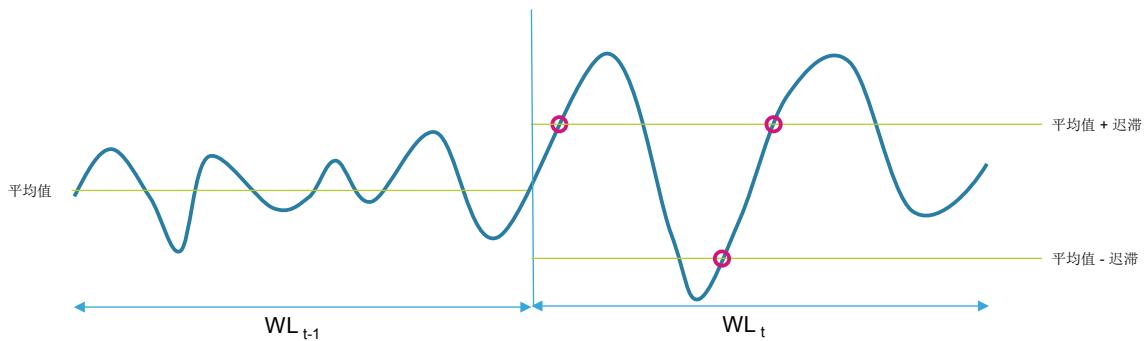


1.3.6

正过零信号

“正过零信号”特征可计算所选输入跨过某个阈值的次数。该内部阈值定义为在前一个窗口中计算的均值（“均值”特征）与用户定义的迟滞之和。此特征仅考虑具有正斜率的过渡。

图 9. 正过零信号

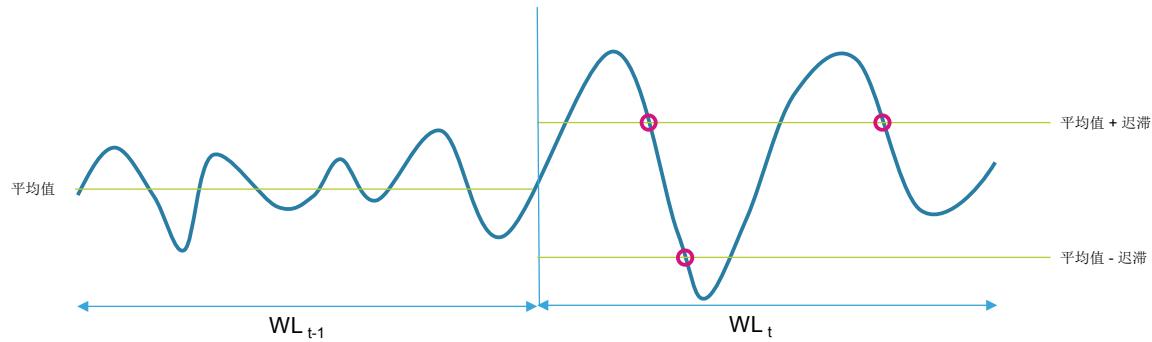


1.3.7

负过零信号

“负过正零信号”特征可计算所选输入跨过某个阈值的次数。该内部阈值定义为在前一个窗口中计算的均值（“均值”特征）与用户定义的迟滞之和。此特征仅考虑具有负斜率的过渡。

图 10. 负过零信号



1.3.8 峰值探测器

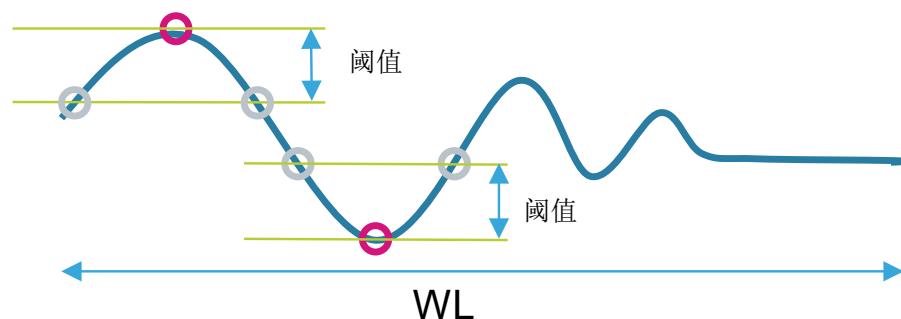
“峰值探测器”特征可计算定义的时间窗口中的所选输入的峰值数（正负）。

用户必须为此特征定义一个阈值，并考虑使用三个值的缓冲区进行评估。如果三值缓冲区的第二个值大于（或小于）所选阈值的其他两个值，则峰值数会增加。

为计算此特征而考虑的三个值的缓冲区是时间窗口内的移动缓冲区。

下图显示了此特征的计算示例，已在时间窗口中检测到两个峰值（一正一负）。

图 11. 峰值探测器



1.3.9 正峰值探测器

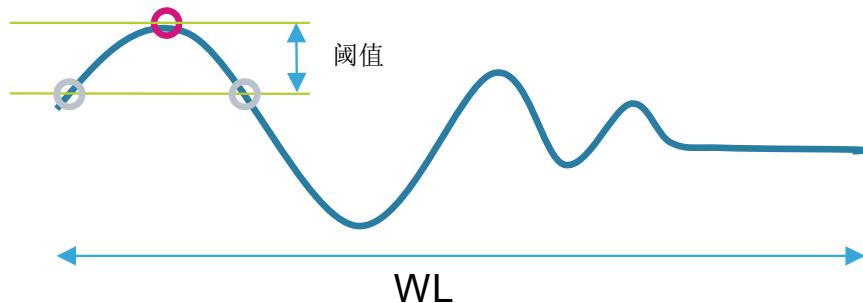
“正峰值探测器”特征可计算定义的时间窗口中的所选输入的正峰值数量。

用户必须为此特征定义一个阈值，并考虑使用三个值的缓冲区进行评估。如果三值缓冲区的第二个值大于所选阈值的其他两个值，则峰值数会增加。

为计算此特征而考虑的三个值的缓冲区是时间窗口内的移动缓冲区。

下图显示了此特征的计算示例，仅在时间窗口中检测到一个峰值（正）。

图 12. 正峰值探测器



1.3.10 负峰值探测器

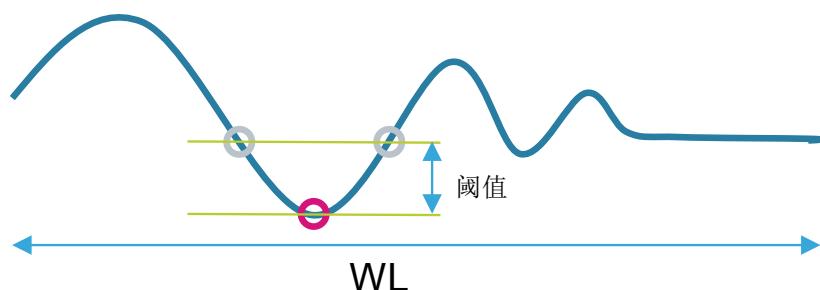
“负峰值探测器”特征可计算定义的时间窗口中的所选输入的负峰值数量。

用户必须为此特征定义一个阈值，并考虑使用三个值的缓冲区进行评估。如果三值缓冲区的第二个值小于所选阈值的其他两个值，则峰值数会增加。

为计算此特征而考虑的三个值的缓冲区是时间窗口内的移动缓冲区。

下图显示了此特征的计算示例，仅在时间窗口中检测到一个峰值（负）。

图 13. 负峰值探测器

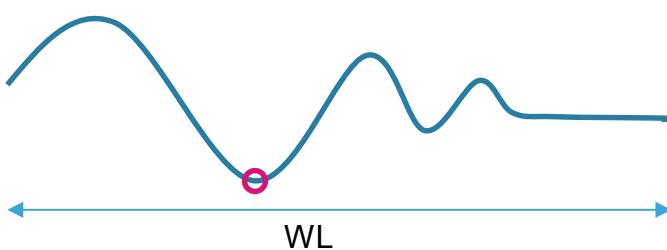


1.3.11 最小值

“最小值”特征可计算的定义的时间窗口中的所选输入的最小值。

下图显示了时间窗口中的最小值示例。

图 14. 最小值

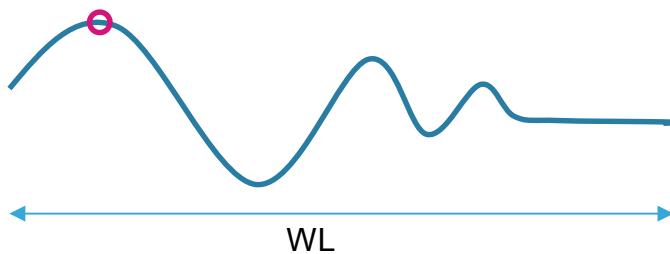


1.3.12 最大值

“最大值”特征可计算定义的时间窗口中的所选输入的最大值。

下图显示了时间窗口中的最大值示例。

图 15. 最大值



1.3.13 特征选择

根据特定应用选择将用于机器学习内核配置的特征。

考虑到使用过多特征可能导致过度拟合，建议通过选择前四个特征来启动：

- 均值
- 方差
- 能量
- 峰到峰

如果这些特征的性能不佳，并且为了提高精度，可以考虑使用其他特征来更好地分离类别。

也必须根据特定应用选择输入（加速度计、陀螺仪）和轴(X, Y, Z, V)。某些类别（如振动）与方向密切相关，因此最好使用独立轴(X, Y, Z)。其他类别（例如行走）与方向无关，因此最好使用范数(V或V2)。

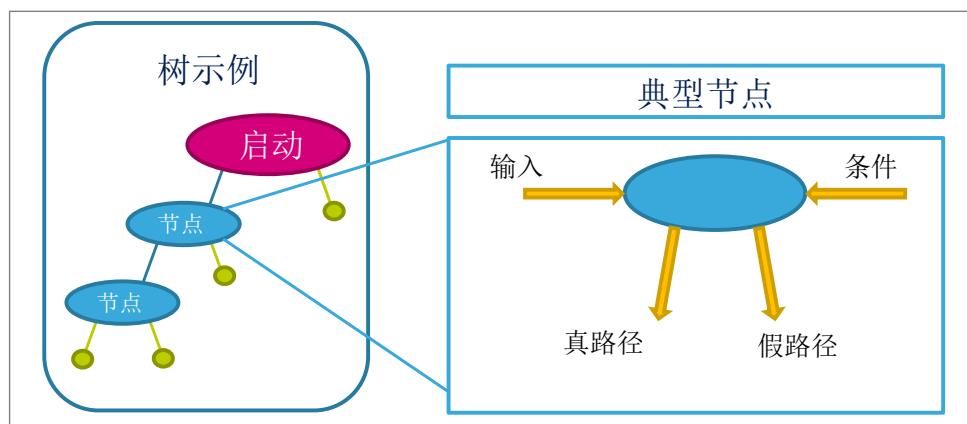
有些，基本特征（均值、方差、能量等...）可能无法帮助区分主导频率，因此可能需要使用滤波器来选择特定频率区域。使用滤波信号可以更精确地区分某些类别。例如，如果用户正在行走，则典型信号约为 1-2 Hz，而如果用户正在慢跑，则典型信号约为 2.5-4 Hz。

1.4 决策树

决策树是根据训练数据构建的预测模型，可以训练数据存储在 LSM6DSOX 中。训练数据是为要识别的每个类别获取的数据日志（在活动识别示例中，类别可能为行走、慢跑、驾驶等）。

前面各节中所述的计算块为决策树的输入。决策树的每个节点都包含一个条件，将在该条件下通过特定阈值来评估特征。如果条件为真，则评估真路径中的下一个节点。如果条件为假，则评估假路径中的下一个节点。决策树的状态将逐节点演化，直至找到结果为止。决策树的结果是在数据收集开始时定义的类别之一。

图 16. 决策树节点



决策树在每个时间窗口（用户为计算特征而设置的“窗口长度”参数）产生一个新结果。也可以通过被称为“元分类器”的附加（可选）滤波器过滤决策树结果，如第 1.5 节 元分类器所述。

可通过 LSM6DSOX 寄存器的嵌入式高级功能页 1 中的专用寄存器访问机器学习内核结果（过滤或未过滤决策树结果）（如表 5 中所示）。可连续读取（轮询）这些寄存器，以检查决策树输出。MLC_STATUS_MAINPAGE (38h) 寄存器包含 8 个可能的决策树的中断状态位。当相应的决策树值更改时，自动将这些位设为 1。此外，也可以通过设置 MLC_INT1 (0Dh) 寄存器将使用这些位生成的中断状态信号驱动到 INT1 引脚，或通过设置 MLC_INT2 (11h) 寄存器驱动至 INT2 引脚（表 6）。当机器学习内核结果改变时，使用中断信号可唤醒执行其他任务或休眠（以节省功耗）的 MCU。

默认情况下，机器学习内核中断信号为脉冲信号。脉冲时长由机器学习内核、有限状态机和传感器 ODR 中的最快 ODR 定义：

```
中断脉冲时长 = 1 / max(MLC_ODR, FSM_ODR, XL_ODR, GYRO_ODR)
```

也可以通过嵌入式功能寄存器 PAGE_RW (17h) 中的 EMB_FUNC_LIR 位将机器学习内核中断信号设为锁存。

表 5. 决策树结果

寄存器	内容
MLC0_SRC (70h)	决策树 1 的结果
MLC1_SRC (71h)	决策树 2 的结果
MLC2_SRC (72h)	决策树 3 的结果
MLC3_SRC (73h)	决策树 4 的结果
MLC4_SRC (74h)	决策树 5 的结果
MLC5_SRC (75h)	决策树 6 的结果
MLC6_SRC (76h)	决策树 7 的结果
MLC7_SRC (77h)	决策树 8 的结果

表 6. 决策树中断

寄存器	内容
MLC_STATUS_MAINPAGE (38h)	将更改的中断状态位包含在决策树结果中
MLC_STATUS (15h)	将更改的中断状态位包含在决策树结果中
MLC_INT1 (0Dh)	允许将决策树的中断状态位发送至 INT1 引脚
MLC_INT2 (11h)	允许将决策树的中断状态位发送至 INT2 引脚

1.4.1

中的决策树限制 LSM6DSOX

就决策树数量、决策树大小、决策树结果的数量而言，LSM6DSOX 的机器学习内核资源有限。

最多可将 8 种不同的决策树存储在 LSM6DSOX 中，但所有决策树的节点总数之和不得超过 256 (*)。LSM6DSOX 中的每个决策树最多可包含 16 个结果。

(*) 该数量可能受所配置的特征和滤波器数量限制。通常，如果使用较少的滤波器和特征，则对决策树的大小没有进一步限制。但是，如果使用许多滤波器和特征，则对决策树的最大节点数略有限制。例如，如果配置的滤波器数量为 10 且配置的特征数为 50，则最大节点数可能会减少 100。该工具用于向用户通知决策树的可用节点。

下表总结了 LSM6DSOX 的限制。

表 7. 中的决策树限制 LSM6DSOX

	LSM6DSOX
最大决策树数量	8
最大节点数（所有决策树的总数）	256 (*)
每个决策树的最大结果数	16

注意：使用多个决策树时，前几节中描述的所有参数（输入、滤波器、在时间窗口中计算的特征、时间窗口本身以及数据速率）对于所有决策树都是通用的。

1.5 元分类器

元分类器是决策树输出滤波器。元分类器使用一些内部计数器来过滤决策树输出。

决策树输出可分为多个子组（例如，可在同一子组中管理相似的类别）。内部计数器可用于决策树输出的所有子组。当决策树的结果是子组类别之一时，特定子组的计数器增加，否则减小。当计数器达到定义的值（被称为“结束计数器”）（由用户设置）时，机器学习内核的输出将更新。

表 8. 元分类器示例

决策树结果	A	A	A	B	A	B	B	B	A	B	B	B	A	A	A
计数器 A (结束计数器 = 3)	1	2	3	2	3	2	1	0	1	0	0	0	1	2	3
计数器 B (结束计数器 = 4)	0	0	0	1	0	1	2	3	2	3	4	5	4	3	2
机器学习内核结果（包括元分类器）	x	x	A	A	A	A	A	A	A	A	B	B	B	B	A

上表显示了通过元分类器过滤决策树输出的效果。该表的第一行包含元分类器之前的决策树输出。计数器 A 和计数器 B 是两个决策树结果（“A”和“B”）的内部计数器。在活动识别示例中，结果“A”可能为行走，结果“B”可能为慢跑。当内部计数器“A”达到值 3（这是计数器“A”的结束计数器）时，将转换为结果“A”。当内部计数器“B”达到值 4 时，将转换为结果“B”。

元分类器的目的为减少假阳性，以避免生成仍不稳定的输出，并减少决策树结果的转换。

1.5.1 中的元分类器限制 LSM6DSOX

元分类器具有数量有限的子组，可在 LSM6DSOX 中使用 4 子组。可能需要使用元分类器对同一子组中的相似类别进行分组。

表 9. 中的元分类器限制 LSM6DSOX

	LSM6DSOX
每个决策树的最大结果数	16
每个决策树的元分类器结果子组	4

注：可配置多个元分类器。一个元分类器可用于机器学习内核中配置的任何决策树。

1.6 有限状态机接口

LSM6DSOX 还提供了一种可配置的有限状态机，该状态机适用于演绎算法，尤其是手势识别。

有限状态机和决策树可组合在一起，以提高运动检测精度。

可通过 LSM6DSOX 中提供的有限状态机检查机器学习内核所产生的决策树结果；这可以通过 CHKDT 条件来实现（如应用笔记 AN5273 LSM6DSOX：有限状态机所述）。

2 机器学习内核工具

通过专用工具可以实现对设备中的机器学习内核的可编程性，该专用工具可作为 Unico GUI 的扩展提供。

2.1 Unico GUI

Unico 是意法半导体产品组合中可用的所有 MEMS 传感器演示板的图形用户界面。它可以与基于 STM32 微控制器（Professional MEMS Tool）的主板进行交互，从而可以在 MEMS 传感器和 PC GUI 之间进行通信。Unico 也可以在未将主板连接到 PC 的情况下离线运行。

可在 www.st.com 上的 STEVAL-MKI109V3 中找到 Professional MEMS Tool 板的详细信息。

Unico GUI 提供支持三种操作系统的三个软件包。

- Windows
 - [STSW-MKI109W](#)
- Linux
 - [STSW-MKI109L](#)
- Mac OS X
 - [STSW-MKI109M](#)

Unico GUI 允许以图形或数字格式显示传感器输出，并允许用户保存或广泛管理来自器件的数据。

通过 Unico 可访问 MEMS 传感器寄存器，从而能够实现寄存器设置的快速原型设计，并可以直接在器件上轻松测试配置。可将当前寄存器配置保存在文本文件中（扩展名为.ucf），并从现有文件中加载配置。这样，就可以在几秒内对传感器重新编程。

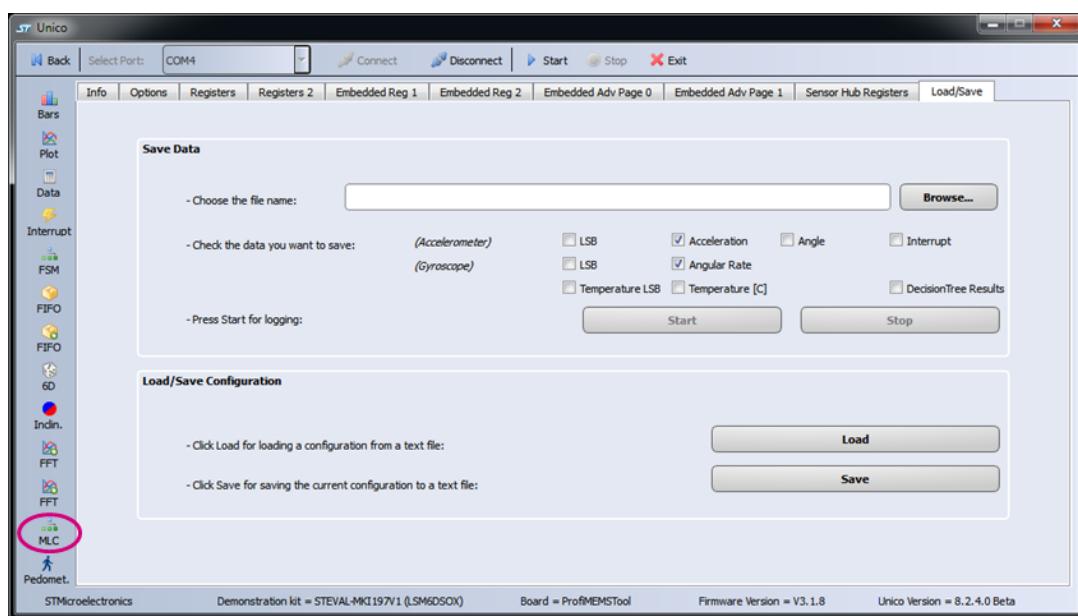
Unico GUI 中的机器学习内核工具通过自动生成设备的配置文件来抽象寄存器配置过程。用户只需在 GUI 中设置一些参数，然后点击几个按钮，便可以使用配置文件。用户可通过这些配置文件创建自己的设备配置库。

机器学习算法需要收集数据日志，可通过 Unico (图 17) 的 Load/Save 标签获取数据日志。对于加速度计，“Acceleration”允许将数据保存为[mg]。对于陀螺仪，“Angular rate”复选框允许将数据保存为[dps]。

注意：记录数据时，必须正确使用“Start”和“Stop”按钮（在 Unico 的 Load/Save 标签中），以免在采集开始或结束时记录错误的数据。例如，在记录“行走”类别的数据模式时，用户应在按下“Start”按钮之前开始行走，并在按下“停 Stop”按钮之后停止行走。

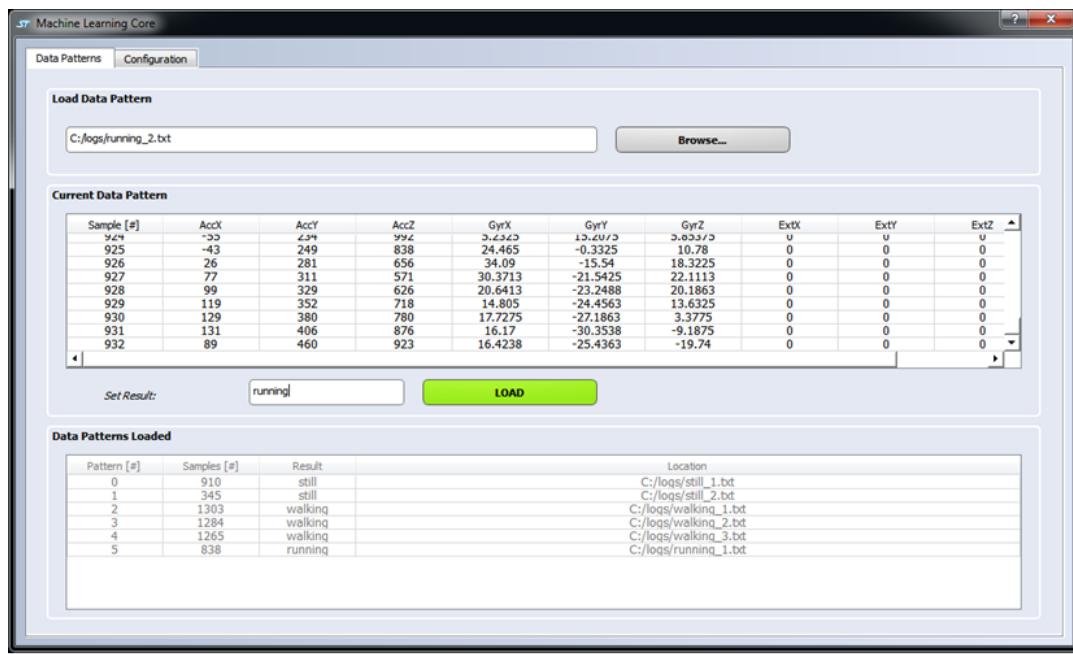
如果离线使用 Unico GUI（未将主板连接到 PC），则已经获取数据日志的用户可直接上传这些日志，以生成机器学习内核配置。

图 17. Unico GUI



然后可以使用“Data Patterns”标签(图 18)将收集到的数据日志载入 GUI 左侧的 Unico 机器学习内核工具中。必须将预期结果分配给每个加载的数据模式（例如，在活动识别算法中，结果可能为：静止、行走、慢跑等...）。此分配也被称为“数据标记”。

图 18. 机器学习内核工具 - Data Patterns



机器学习内核工具的数据模式选项卡中的预期数据测量单位为：

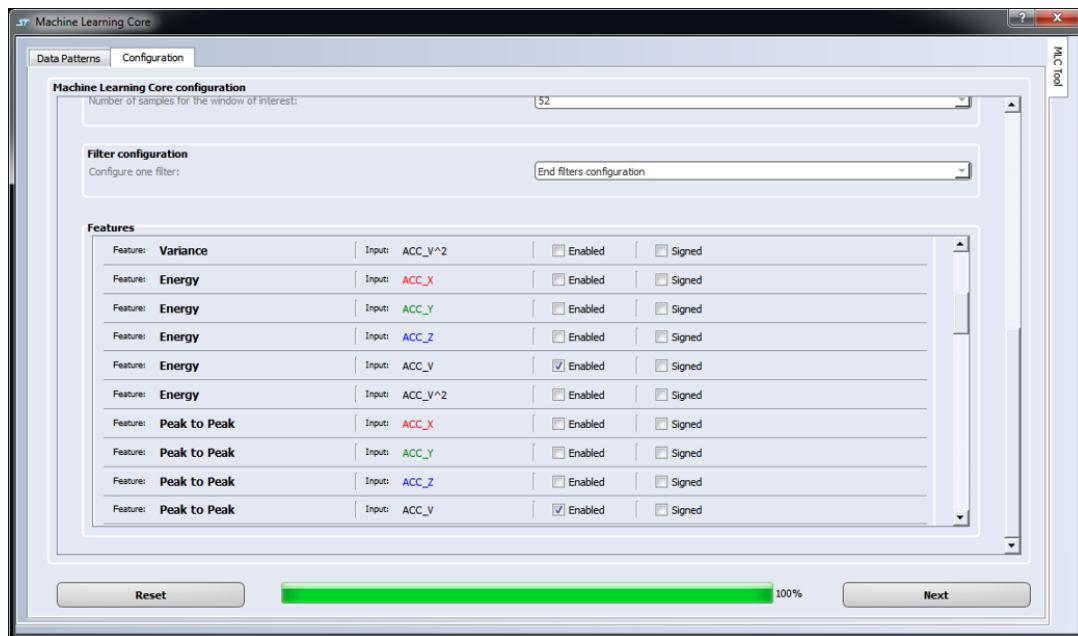
- 加速度计为[mg]
- 陀螺仪为[dps]

对于加速度计，从[mg]转换为[g]，对于陀螺仪，从[dps]转换为[rad/s]，并由机器学习内核工具从内部自动管理这些转换，以使机器学习内核逻辑能够处理正确的数据（[g]和[rad/s]）。对于外部传感器数据，需要用户在配置的后期阶段设置适当的灵敏度。

在机器学习内核工具(图 19)的“Configuration”标签中，可配置机器学习内核的所有参数（如 ODR、满量程、窗口长度、滤波器、特征、元分类器）。该工具允许选择可应用于原始数据的多个滤波器，并通过输入数据或滤波数据计算多个特征。计算出的特征将成为决策树的属性。

当已连接并配置好设备后，该工具将根据当前的设备配置自动建议 ODR 和满量程（对于加速度计和陀螺仪）。

图 19. 机器学习内核工具 - 配置



机器学习内核工具的“Configuration”标签将生成属性关系文件(ARFF)，这是决策树生成过程的起点。决策树可由不同的机器学习工具生成([第 2.2 节](#))。

决策树生成后，可将其上传到 Unico 中的机器学习内核工具，以完成 LSM6DSOX 的寄存器配置生成。

通过访问传感器寄存器，Unico GUI 可读取决策树输出状态，使其与传感器数据一起显示，并可以将所有数据（传感器输出和决策树输出）一起记录在同一文本文件中。

2.2 决策树生成

几种机器学习工具能够生成决策树。最常用的工具之一为怀卡托大学开发的 Weka 软件（有关此软件的更多详细信息，请参见[附录 A](#)）。其他替代工具包括：[RapidMiner](#) ([附录 B](#))、[Matlab](#) ([附录 C](#))、[Python](#) ([附录 D](#))。

Weka 能够从属性关系文件（ARFF）生成决策树。通过 Weka 可以评估哪些属性适合决策树，并且可以通过更改 Weka 中的所有可用参数来实现不同的决策树配置。图 20 图 21 显示了“Preprocess”和“Classify”选项卡，这些选项卡可用于评估属性并生成决策树。

图 20. Weka Preprocess

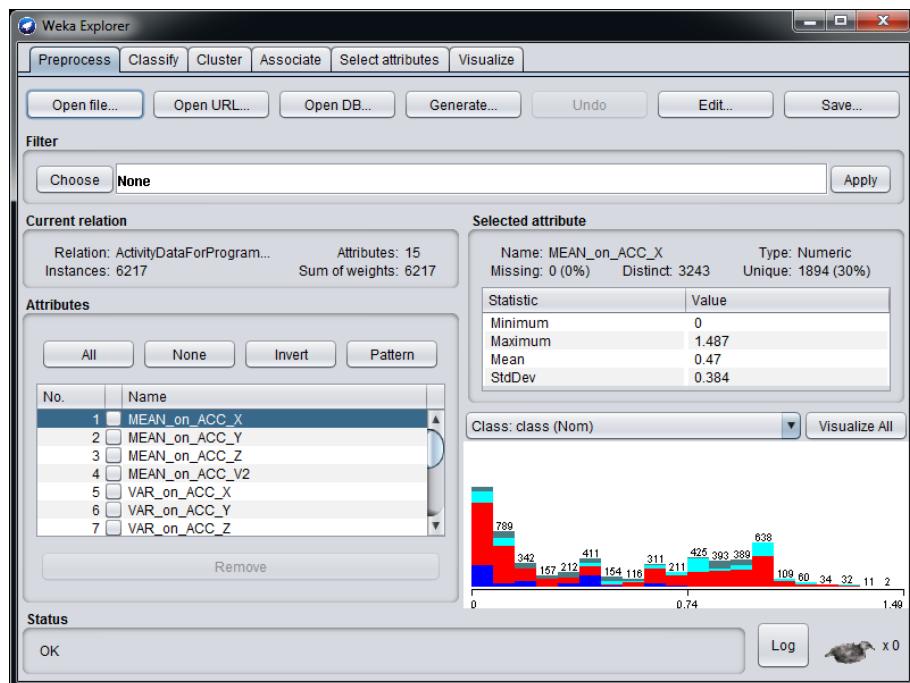
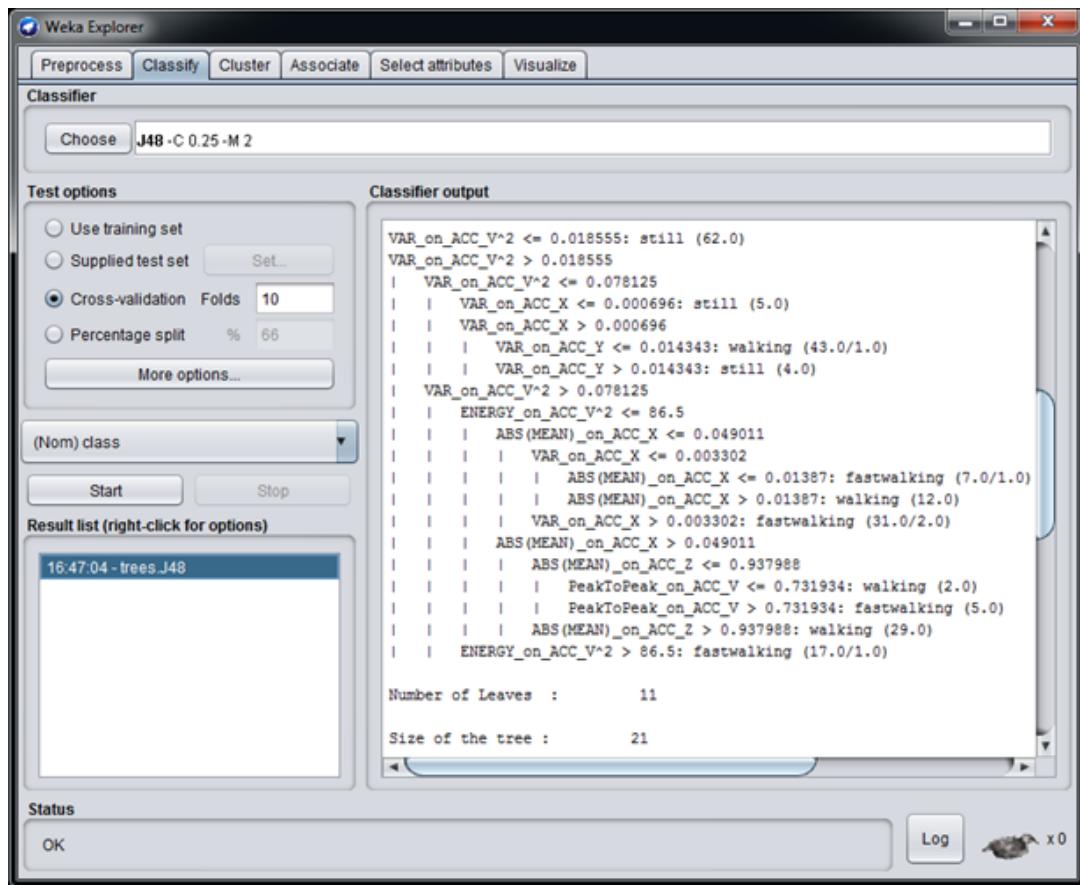


图 21. Weka Classify



决策树生成后，可将其上传到 Unico 中的机器学习内核工具，以完成 LSM6DSOX 的寄存器配置生成。

Unico 中的机器学习内核工具允许以文本格式(.txt)输入决策树文件。文本文件必须包含采用 Weka J48 格式的决策树（决策树示例如图 22 中所示）。在 Weka 分类器输出（图 21）中，必须从第一行（第一个节点）选择决策树，或选择 RapidMiner 格式（附录 B）的决策树。最后两行（叶数和树的大小）可选。必须将 Weka 的选定输出复制到文本文件中。

图 22. 决策树格式

```

example_DcisionTree_format.txt - Notepad
File Edit Format View Help
VAR_on_ACC_V^2 <= 0.018555: still (62.0)
VAR_on_ACC_V^2 > 0.018555
    VAR_on_ACC_V^2 <= 0.078125
        VAR_on_ACC_X <= 0.000696: still (5.0)
        VAR_on_ACC_X > 0.000696
            VAR_on_ACC_Y <= 0.014343: walking (43.0/1.0)
            VAR_on_ACC_Y > 0.014343: still (4.0)
    VAR_on_ACC_V^2 > 0.078125
        ENERGY_on_ACC_V^2 <= 86.5
            ABS(MEAN)_on_ACC_X <= 0.049011
                VAR_on_ACC_X <= 0.003302
                    ABS(MEAN)_on_ACC_X <= 0.01387: fastwalking (7.0/1.0)
                    ABS(MEAN)_on_ACC_X > 0.01387: walking (12.0)
                VAR_on_ACC_X > 0.003302: fastwalking (31.0/2.0)
            ABS(MEAN)_on_ACC_X > 0.049011
                ABS(MEAN)_on_ACC_Z <= 0.937988
                    PeakToPeak_on_ACC_V <= 0.731934: walking (2.0)
                    PeakToPeak_on_ACC_V > 0.731934: fastwalking (5.0)
                ABS(MEAN)_on_ACC_Z > 0.937988: walking (29.0)
        ENERGY_on_ACC_V^2 > 86.5: fastwalking (17.0/1.0)

Number of Leaves : 11
Size of the tree : 21

```

如果决策树是由其他工具生成，则必须将格式转换为 Weka J48 格式（或 RapidMiner 格式），以使 Unico 中的机器学习内核工具能够正确读取决策树。

2.3 配置过程

图 23 显示从数据模式到生成器件（LSM6DSOX）的寄存器设置的整个机器学习处理过程。

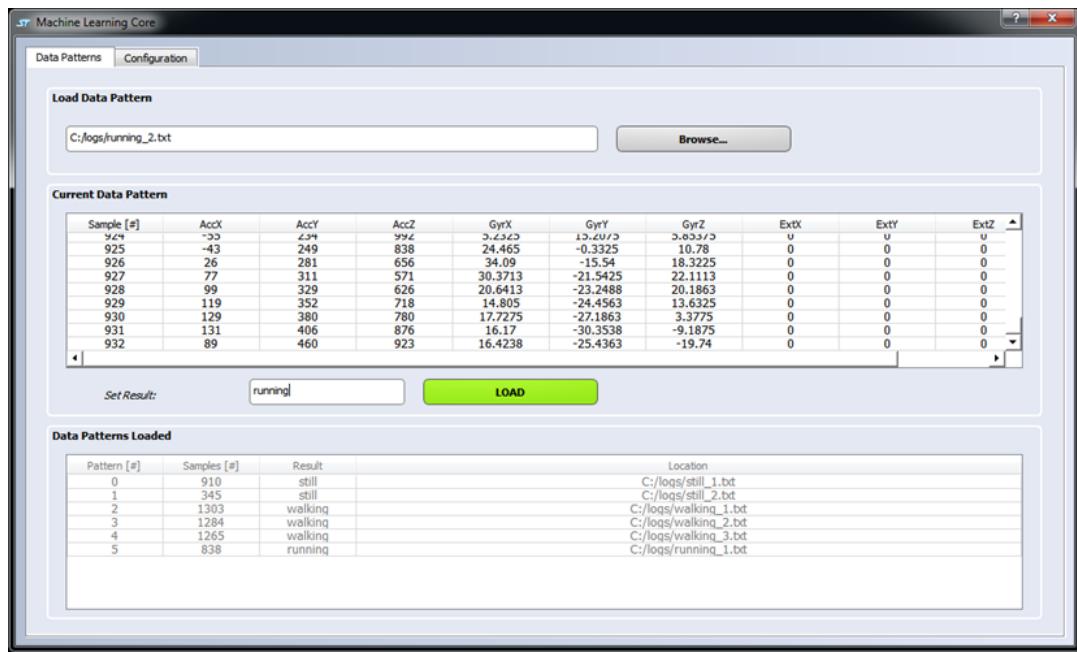
如第 2.1 节 Unico GUI 中所示，可通过 Unico GUI 的“Load/Save”选项卡获取数据模式。如果该操作无法实现，或如果用户希望使用一些不同的数据模式，用户仍可以将这些数据模式上传到 Unico 的机器学习内核工具中，但有一些限制：

- 每种数据模式均必须以标题行开头，其中包含数据测量单位
 - A_X [mg] A_Y [mg] A_Z [mg] G_X [dps] G_Y [dps] G_Z [dps]
- 标题行之后的数据必须用“制表符”或“空格”分隔。
- 文件列中的传感器顺序必须为加速度计数据（如果有）、陀螺仪数据（如果有）、外部传感器数据（如果有）。
- 任何传感器列中的轴顺序为 X、Y、Z。



打开 Unico 中的机器学习内核工具，可加载以上述格式获取的数据模式，并为每个数据日志分配预期结果（如下图所示）。

图 24. 将结果分配给数据模式



加载所有数据模式后，可通过配置选项卡配置机器学习内核参数。这些参数为 ODR、满量程、决策树数量、窗口长度、滤波器、特征等...（如图 25、图 26、图 27、图 28 中所示）。

图 25. 机器学习内核配置

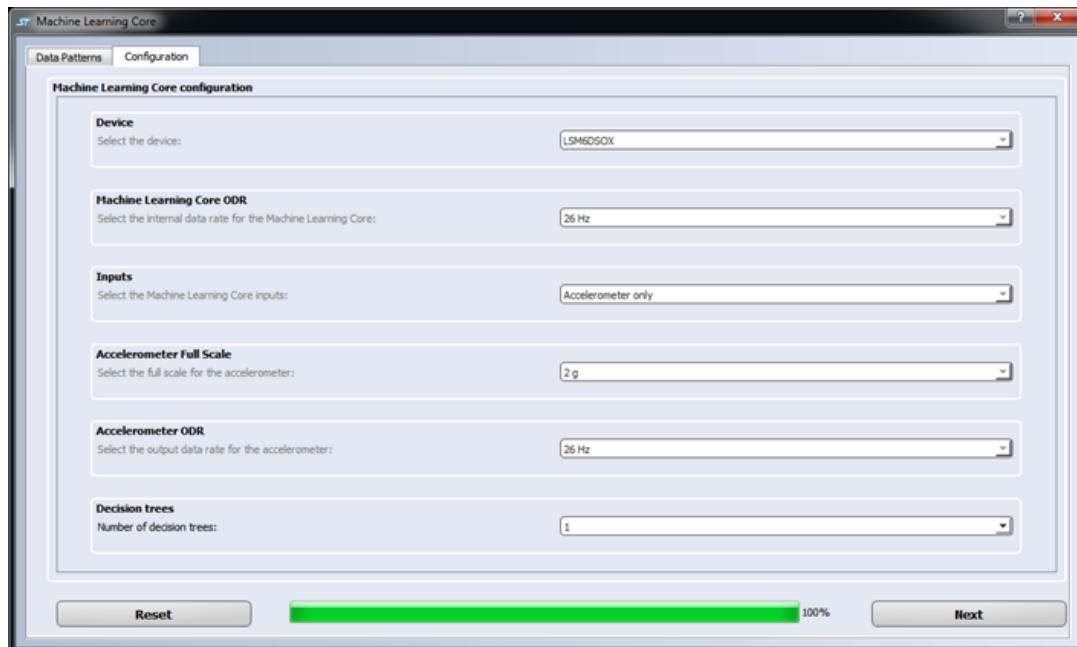


图 26. 滤波器配置

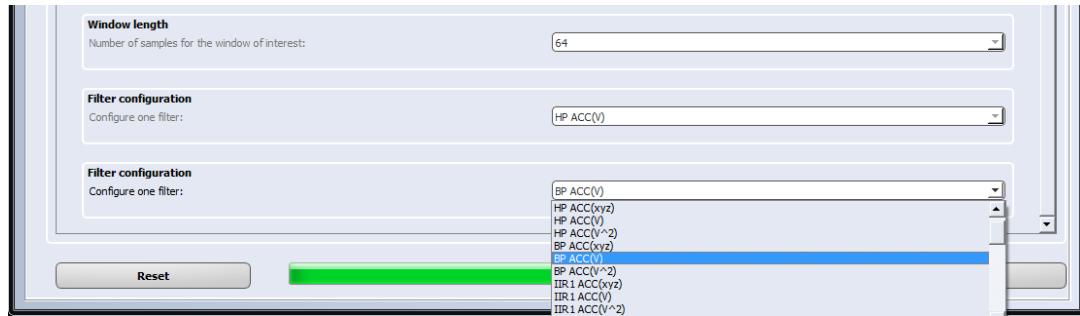


图 27. 特征配置

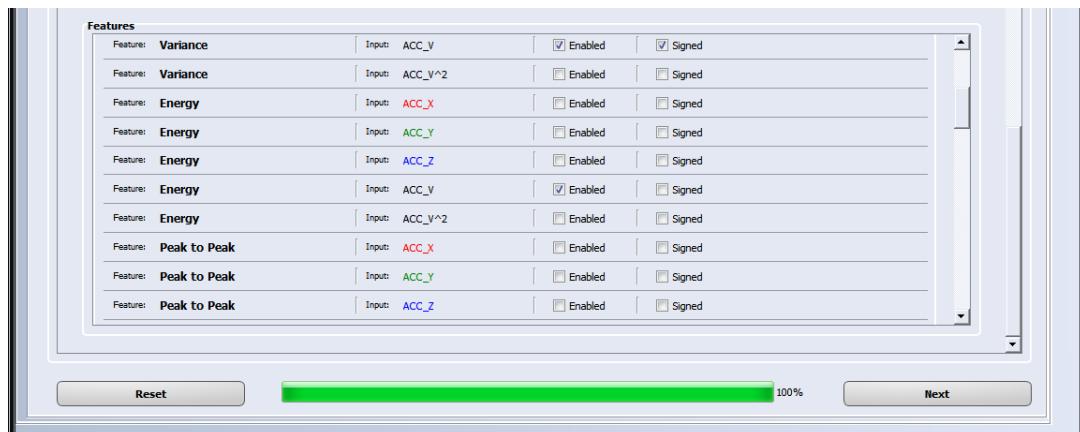
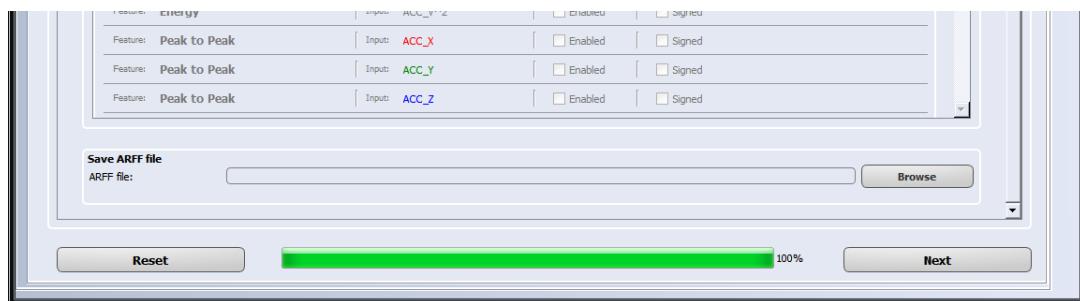


图 28. ARFF 生成



可选择多个滤波器和多个特征。在选择“End filter configuration”参数之前，GUI 会反复请求另一个滤波器(图 26)。可使用复选框轻松选择所有可用特征(图 27)。

配置完所有特征后，Unico 中的机器学习内核工具将生成一个 ARFF 文件（图 28），该文件包含根据训练数据计算出的所有特征。图 29 显示了 Unico 中的机器学习内核工具生成的 ARFF 文件示例。

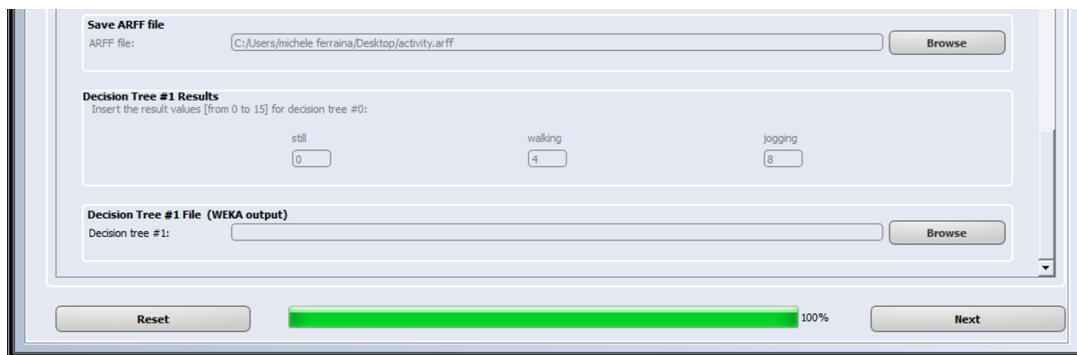
可将生成的 ARFF 文件加载到 Weka 中，以构建决策树。如果不使用 Weka 构建决策树，用户必须将 ARFF 文件转换为任何其他决策树生成工具所需的文件格式。在该特定情况下，可能还需要决策树格式转换为第 2.2 节 决策树生成中所述的 Weka J48 格式。

图 29. ARFF 文件

```
1 #relation 'WEC'
2
3 @attribute ABS(MEAN)_on_ACC_X numeric
4 @attribute ABS(MEAN)_on_ACC_Y numeric
5 @attribute ABS(MEAN)_on_ACC_Z numeric
6 @attribute ABS(MEAN)_on_ACC_V^2 numeric
7 @attribute ABS(VAR)_on_ACC_X numeric
8 @attribute ABS(VAR)_on_ACC_Y numeric
9 @attribute ABS(VAR)_on_ACC_Z numeric
10 @attribute ABS(VAR)_on_ACC_V^2 numeric
11 @attribute ABS(ENERGY)_on_ACC_V^2 numeric
12 @attribute ABS(ENERGY)_on_filter_BP_on_ACC_V^2 numeric
13 @attribute ABS(ENERGY)_on_filter_IIR2_on_ACC_V^2 numeric
14 @attribute ABS(zeroCross)_on_filter_BP_on_ACC_V^2 numeric
15 @attribute ABS(zeroCross)_on_filter_IIR2_on_ACC_V^2 numeric
16 @attribute ABS(PeakToPeak)_on_ACC_V numeric
17 @attribute class {Stationary, Walking, Jogging, Biking}
18
19 #data
20 0.0058403, 0.0390625, 1.01465, 1.03906, 4.17233e-06, 3.8147e-06, 0.00683594, 0.0107422, 80.25, 0.585938, 73.75, 0, 1, 0.0144484, Stationary
21 0.0059697, 0.0389404, 1.01465, 1.03613, 2.02656e-06, 7.62939e-06, 0.00683594, 0.0087906, 79.875, 0.0004704, 79.9375, 0, 0, 0.0107422, Stationary
22 0.00609207, 0.039978, 1.01465, 1.03213, 4.44916e-06, 4.76837e-06, 0.00976562, 0.00488281, 80.375, 0.00071523, 80.625, 0, 0, 0.0136719, Stationary
23 0.00559235, 0.0400696, 1.0127, 1.03516, 3.03984e-06, 7.62939e-06, 0.0078125, 0.00585938, 80, 0, 0.00043273, 80, 0, 0, 0.0126953, Stationary
24 0.005518744, 0.0392151, 1.01562, 1.03418, 3.45707e-06, 8.58307e-06, 0.00976562, 0.010436101, 80.0625, 0, 0, 0.0144484, Stationary
25 0.00550624, 0.0394592, 1.0166, 1.0332, 4.05312e-06, 9.53674e-06, 0.00195312, 0.000976562, 80.1875, 0.00039053, 80.125, 0, 0, 0.0126953, Stationary
26 0.00591278, 0.0392761, 1.0166, 1.03418, 3.69549e-06, 7.62939e-06, 0.00976562, 0, 0.025, 0.000770569, 80.9375, 0, 0, 0.0126953, Stationary
27 0.00541306, 0.0393677, 1.01465, 1.03125, 3.27282e-06, 1.81198e-05, 0, 0.000976562, 79.875, 0.000643757, 80, 0, 0, 0.0136719, Stationary
28 0.00675964, 0.0389099, 1.01562, 1.03613, 1.90735e-06, 1.14441e-05, 0.00292969, 0.00292969, 80.3125, 0.000609875, 80.5, 0, 0, 0.0107422, Stationary
29 0.00554276, 0.0393066, 1.01465, 1.03125, 3.57622e-06, 6.67572e-06, 0.00976562, 0.000976562, 79.9375, 0.000419617, 80.0625, 0, 0, 0.0107422, Stationary
30 0.00581741, 0.0394287, 1.01367, 1.03223, 6.19888e-06, 1.04904e-05, 0.00292969, 0.00195312, 80.125, 0.00067586, 80.5, 0, 0, 0.0126953, Stationary
31 0.006045392, 0.0397339, 1.0127, 1.03613, 2.50304e-06, 1.04904e-05, 0.00976562, 0.00683594, 80.0625, 0.0004499725, 80, 0, 0, 0.0107422, Stationary
32 0.00537109, 0.039093, 1.0127, 1.03613, 2.86102e-06, 6.67572e-06, 0.00878906, 0.00585938, 80.125, 0.000460425, 80.0425, 0, 0, 0.0136719, Stationary
33 0.00451169, 0.0394592, 1.01465, 1.0332, 2.26498e-06, 1.52588e-06, 0.00292969, 0.00195312, 80.25, 0, 0, 0.0117188, Stationary
34 0.00587463, 0.0387573, 1.01367, 1.0332, 3.63508e-06, 9.53674e-06, 0.00976562, 0.00292969, 80.3125, 0.000118351, 80.875, 0, 0, 0.0175781, Stationary
35 0.006146377, 0.0388794, 1.01367, 1.03613, 3.99351e-06, 7.62939e-06, 0.00488281, 0.00878906, 79.9375, 0.000247002, 80.875, 0, 0, 0.0107422, Stationary
36 0.006430569, 0.0393066, 1.01465, 1.03223, 3.15905e-06, 8.58307e-06, 0.00976562, 0.00390425, 80.25, 0.000587003, 80.1875, 0, 0, 0.0126953, Stationary
37 0.0174406, 0.014163, 1.02051, 1.04199, 4.4373e-06, 2.86102e-06, 0, 0.00585938, 81.0625, 0.000494295, 81.4875, 0, 0, 0.0126953, Stationary
38 0.01855497, 0.014163, 1.01953, 1.04001, 7.86781e-06, 1.90735e-06, 0, 0.00047002, 81.0625, 0, 0, 0.0117188, Stationary
39 0.01855497, 0.01362, 1.01855, 1.03711, 5.77056e-06, 4.19735e-07, 0, 0, 0.00047002, 81.0625, 0, 0, 0.0126953, Stationary
40 0.01853105, 0.0145569, 1.01953, 1.03906, 5.18565e-06, 6.66939e-06, 0.000976562, 0.00292969, 80.875, 0.00050679, 81.5425, 0, 0, 0.0117188, Stationary
41 0.01853105, 0.0145569, 1.01953, 1.03906, 5.18565e-06, 6.66939e-06, 0.000976562, 0.00292969, 80.875, 0.00050679, 81.5425, 0, 0, 0.0117188, Stationary
42 0.01853105, 0.0145569, 1.01953, 1.03906, 5.18565e-06, 6.66939e-06, 0.000976562, 0.00292969, 80.875, 0.00050679, 81.5425, 0, 0, 0.0117188, Stationary
43 0.01853105, 0.0145569, 1.01953, 1.03906, 5.18565e-06, 6.66939e-06, 0.000976562, 0.00292969, 80.875, 0.00050679, 81.5425, 0, 0, 0.0117188, Stationary
44 0.01853105, 0.0145569, 1.01953, 1.03906, 5.18565e-06, 6.66939e-06, 0.000976562, 0.00292969, 80.875, 0.00050679, 81.5425, 0, 0, 0.0117188, Stationary
45 0.0175171, 0.0140076, 1.01953, 1.04199, 6.4373e-06, 2.98023e-06, 0.00292969, 0.00488281, 81.1875, 0, 0, 0.0136719, Stationary
```

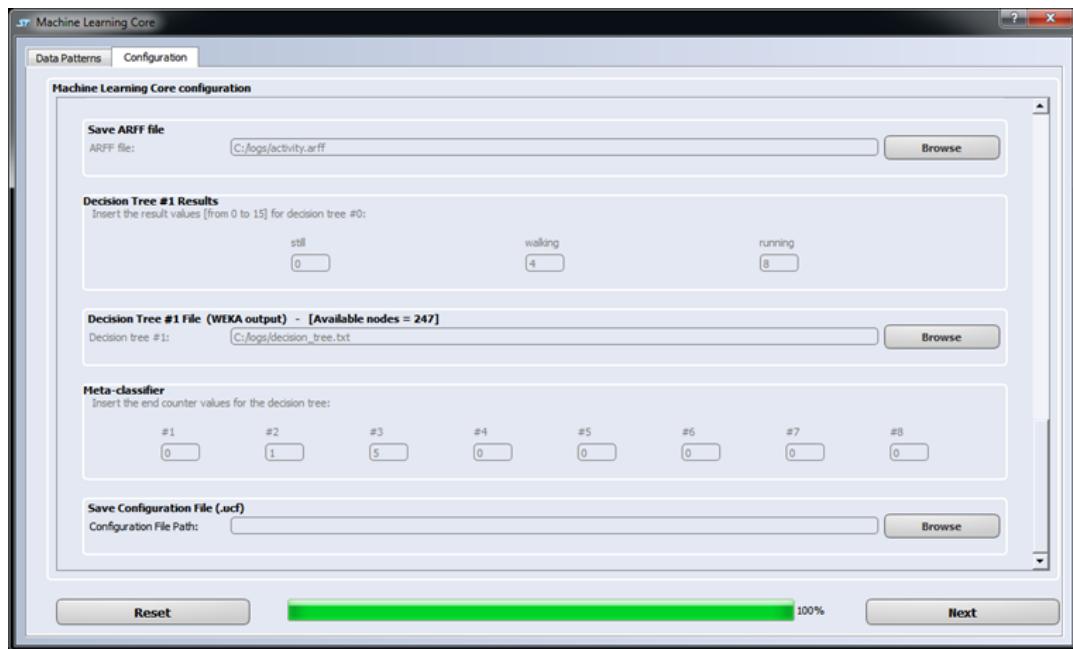
在为决策树配置结果值之后，可以将决策树（采用 Weka J48 格式）加载到 Unico GUI 的机器学习内核工具中，以完成配置并获取器件（LSM6DSOX）的寄存器设置。

图 30. 结果和决策树配置



配置过程的最后一步为配置元分类器，这是用于生成决策树结果的可选滤波器。之后，工具已准备好生成（图 30）器件的配置。

图 31. 元分类器和设备配置



当器件的寄存器配置已保存时，可使用 Unico GUI 的“Load/Save”选项卡将其加载到器件中。

图 32. Unico 载入配置



对器件进行编程后，可在 Unico (图 33)的数据窗口中或在包含机器学习内核源寄存器(图 34)的其中一个寄存器选项卡中监控机器内核学习结果。

图 33. Unico 数据窗口



图 34. Unico - 机器学习内核源寄存器

MLC0_SRC	(70h)	01	Read	Write	Default
MLC1_SRC	(71h)	00	Read	Write	Default
MLC2_SRC	(72h)	00	Read	Write	Default
MLC3_SRC	(73h)	00	Read	Write	Default
MLC4_SRC	(74h)	00	Read	Write	Default
MLC5_SRC	(75h)	00	Read	Write	Default
MLC6_SRC	(76h)	00	Read	Write	Default
MLC7_SRC	(77h)	00	Read	Write	Default

3 决策树示例

本节介绍了可载入 LSM6DSOX 中的决策树的一些示例。

3.1 振动监测

下图中的决策树显示了简单的振动监测示例。使用只有一个特征的简单决策树即可识别三种不同的振动级别（振动 1、振动 2、振动 3），该特征为加速度计范数平方输入(p2p_accNorm2)中的峰峰值特征。

该振动监测示例以 26 Hz 运行，并在具有 16 个样本的窗口中计算特征。LSM6DSOX 在 1.8 V 下的电流消耗约为 171 μ A。关闭机器学习内核时，LSM6DSOX 的电流消耗约为 170 μ A，因此机器学习内核的额外电流消耗仅为 1 μ A。

图 35. 振动监测决策树

```
DecisionTreeVibration.txt - Notepad
File Edit Format View Help
p2p_accNorm2 <= 0.03: vibration1
p2p_accNorm2 > 0.03
|   p2p_accNorm2 <= 1.5: vibration2
|   p2p_accNorm2 > 1.5: vibration3

Number of Leaves : 3
Size of the tree : 5
```

3.2 运动强度

下图中的决策树显示了仅使用加速度计范数中的“方差”特征所实现的运动强度的简单示例。该决策树可识别八种不同的强度级别。

本示例中所述的运动强度配置以 12.5 Hz 运行，并在具有 39 个样本的窗口中计算特征。LSM6DSOX 在 1.8 V 下的电流消耗约为 171 μ A。关闭可编程传感器时，LSM6DSOX 的电流消耗约为 170 μ A，因此机器学习内核的额外电流消耗仅为 1 μ A。

图 36. 运动强度决策树

```

dec_tree.txt - Notepad
File Edit Format View Help
module_variance <= 0.009: Intensity_0
module_variance > 0.009
|   module_variance <= 0.013671875: Intensity_1
|   module_variance > 0.013671875
|       module_variance <= 0.0234375: Intensity_2
|       module_variance > 0.0234375
|           module_variance <= 0.033203125: Intensity_3
|           module_variance > 0.033203125
|               module_variance <= 0.078125: Intensity_4
|               module_variance > 0.078125
|                   module_variance <= 0.1640625: Intensity_5
|                   module_variance > 0.1640625
|                       module_variance <= 0.3125: Intensity_6
|                       module_variance > 0.3125: Intensity_7
Number of Leaves : 8
Size of the tree : 15

```

3.3 6D 位置识别

LSM6DSOX 已将 6D 位置识别算法嵌入到设备中。本节中所述的示例仅显示了使用决策树的不同实现。使用以下特征，可通过简单的决策树(图 38)轻松识别六种不同的位置(图 37):

- *meanx_abs*: 加速度计 X 轴的均值（无符号）
- *meany_abs*: 加速度计 Y 轴的均值（无符号）
- *meanz_abs*: 加速度计 Z 轴的均值（无符号）
- *meanx_s*: 加速度计 X 轴的均值（带符号）
- *meany_s*: 加速度计 Y 轴的均值（带符号）
- *meanz_s*: 加速度计 Z 轴的均值（带符号）

本示例中所述的 6D 位置识别配置以 26 Hz 运行，并在具有 16 个样本的窗口中计算特征。LSM6DSOX 在 1.8 V 下的电流消耗约为 172 μ A。关闭机器学习内核时，LSM6DSOX 的电流消耗约为 170 μ A，因此机器学习内核的额外电流消耗仅为 2 μ A。

图 37. 6D 位置

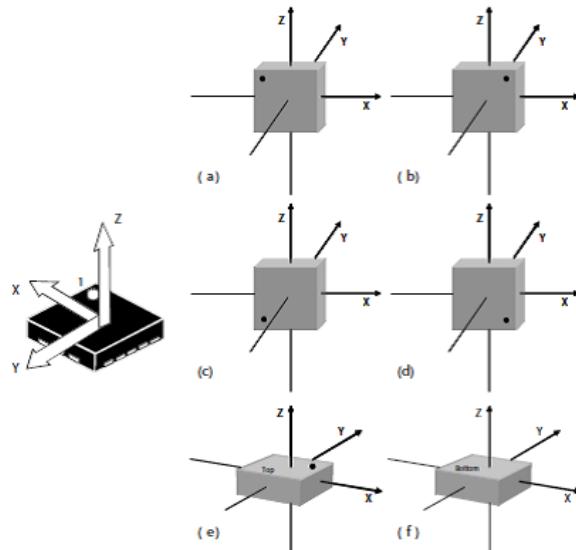


图 38. 6D 决策树

```
six_d.txt - Notepad
File Edit Format View Help
meanx_abs <= 0.3
|   meany_abs <= 0.3
|   |   meanz_s <= 0.3: zdw
|   |   meanz_s > 0.3: yup
|   meany_abs > 0.3
|   |   meanz_abs <= 0.3
|   |   |   meany_s <= 0.3: ydw
|   |   |   meany_s > 0.3: yup
|   |   meanz_abs > 0.3 : others
meanx_abs > 0.3
|   meanz_abs <= 0.3
|   |   meany_abs <= 0.3
|   |   |   meanx_s <= 0.3 : xdw
|   |   |   meanx_s > 0.3: yup
|   |   meany_abs > 0.3 : others
|   meanz_abs > 0.3: others

Number of Leaves : 9
Size of the tree : 17
```

3.4

智能手机应用程序的活动识别

本示例中所述的活动识别算法旨在用于智能手机应用程序，因为为此目的收集的所有数据日志都是通过用户口袋中的智能手机获取的。已从不同的人那里获得数以百计的数据日志，由于不同的人以不同的方式行走或奔跑，从而增加了算法的复杂度。

已选择所有可能活动的小子集，以提高识别算法的精度。本示例中识别的活动子集为：静止、行走、慢跑和骑自行车。

已使用四个特征（均值、方差、峰到峰、过零信号），并且两种不同的滤波器已应用于加速度计输入数据。下表显示了活动识别配置。

表 10. 智能手机配置的活动识别

配置	加速度计，26 Hz ODR，4 g 满量程
窗口长度	75 个样本（约 3 秒）
滤波器	加速度计范数上的带通 加速度计范数平方上的 IIR2
特征	均值 方差 峰到峰 过零信号
输出	静止（0） 行走（1） 慢跑（4） 骑自行车（8）
元分类器	静止和行走时为 0 慢跑时为 1 骑自行车时为 4

图 39 显示 Weka 所生成的决策树。Weka (图 40) 的交叉验证结果表明，正确分类的实例占 96.7%。

活动识别示例的配置以 26 Hz 运行，在具有 75 个样本的窗口中计算特征。LSM6DSOX 在 1.8 V 下的电流消耗约为 174 μ A。关闭机器学习内核时，LSM6DSOX 的电流消耗约为 170 μ A，因此机器学习内核的额外电流消耗仅为 4 μ A。

图 39. 智能手机决策树的活动识别

```

C:\Users\michele.ferrina\Documents\ProgrammableSensor\Configurations_MPLSM6DSO\ActivityRecognition_forMobile\v0.7_good\decree.txt - Notepad++ [Administrator]

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? 

decree.txt

1 ABS(VAR)_on_ACC_V-2 <= 0.032227
2 | ABS(VAR)_on_ACC_V-2 <= 0.012695
3 | | ABS(ENERGY)_on_filter_BP_on_ACC_V-2 <= 0.023254: Stationary (6920.0/17.0)
4 | | ABS(ENERGY)_on_filter_BP_on_ACC_V-2 > 0.023254
5 | | ABS(MEAN)_on_ACC_V-2 <= 0.944824
6 | | | ABS(ENERGY)_on_filter_IIR2_on_ACC_V-2 <= 64.8125: Biking (26.0)
7 | | | ABS(ENERGY)_on_filter_IIR2_on_ACC_V-2 > 64.8125: Walking (8.0)
8 | | | ABS(MEAN)_on_ACC_V-2 > 0.944824
9 | | | ABS(ZeroCross)_on_filter_IIR2_on_ACC_V-2 <= 11: Stationary (2393.0/103.0)
10 | | | ABS(ZeroCross)_on_filter_IIR2_on_ACC_V-2 > 11
11 | | | | ABS(MEAN)_on_ACC_V-2 <= 0.999023: Walking (13.0)
12 | | | | ABS(MEAN)_on_ACC_V-2 > 0.999023
13 | | | | ABS(VAR)_on_ACC_V-2 <= 0.01123: Stationary (27.0/1.0)
14 | | | | ABS(VAR)_on_ACC_V-2 > 0.01123: Biking (6.0)
15 ABS(VAR)_on_ACC_V-2 > 0.012695
16 | ABS(ENERGY)_on_filter_IIR2_on_ACC_V-2 <= 75.125
17 | | ABS(ZeroCross)_on_filter_IIR2_on_ACC_V-2 <= 8
18 | | | ABS(MEAN)_on_ACC_V-2 <= 0.95752: Walking (25.0/2.0)
19 | | | ABS(MEAN)_on_ACC_V-2 > 0.95752
20 | | | ABS(PeakToPeak)_on_ACC_V <= 0.371094
21 | | | | ABS(VAR)_on_ACC_V-2 <= 0.016113
22 | | | | ABS(ZeroCross)_on_filter_IIR2_on_ACC_V-2 <= 7: Stationary (8.0/1.0)
23 | | | | ABS(ZeroCross)_on_filter_IIR2_on_ACC_V-2 > 7: Biking (6.0)
24 | | | | ABS(VAR)_on_ACC_V-2 > 0.016113: Biking (32.0)
25 | | | | ABS(PeakToPeak)_on_ACC_V > 0.371094
26 | | | | ABS(VAR)_on_ACC_V-2 <= 0.025391: Stationary (33.0/1.0)
27 | | | | ABS(VAR)_on_ACC_V-2 > 0.025391
28 | | | | ABS(MEAN)_on_ACC_V-2 <= 0.977539: Stationary (7.0/3.0)
29 | | | | ABS(MEAN)_on_ACC_V-2 > 0.977539: Biking (16.0)
30 ABS(ZeroCross)_on_filter_IIR2_on_ACC_V-2 > 8
31 | ABS(ENERGY)_on_filter_BP_on_ACC_V-2 <= 0.427734: Walking (152.0/5.0)
32 | ABS(ENERGY)_on_filter_BP_on_ACC_V-2 > 0.427734
33 | | ABS(ENERGY)_on_ACC_V-2 <= 74: Biking (22.0/1.0)
34 | | ABS(ENERGY)_on_ACC_V-2 > 74
35 | | ABS(ENERGY)_on_filter_BP_on_ACC_V-2 <= 0.525391: Walking (9.0)
36 | | ABS(ENERGY)_on_filter_BP_on_ACC_V-2 > 0.525391: Stationary (7.0/3.0)
37 ABS(ENERGY)_on_filter_IIR2_on_ACC_V-2 > 75.125
38 | ABS(MEAN)_on_ACC_V-2 <= 1.13867
39 | | ABS(ENERGY)_on_filter_BP_on_ACC_V-2 <= 0.072449: Stationary (61.0)
40 | | ABS(ENERGY)_on_filter_BP_on_ACC_V-2 > 0.072449
41 | | | ABS(VAR)_on_ACC_V-2 <= 0.021973
42 | | | | ABS(MEAN)_on_ACC_V-2 <= 0.993164: Biking (13.0/1.0)
43 | | | | ABS(MEAN)_on_ACC_V-2 > 0.993164
44 | | | | ABS(ENERGY)_on_filter_IIR2_on_ACC_V-2 <= 90.625
45 | | | | ABS(ZeroCross)_on_filter_IIR2_on_ACC_V-2 <= 5
46 | | | | ABS(VAR)_on_ACC_V-2 <= 0.018555
47 | | | | ABS(ENERGY)_on_filter_BP_on_ACC_V-2 <= 0.079529: Biking (5.0)
48 | | | | ABS(ENERGY)_on_filter_BP_on_ACC_V-2 > 0.079529
49 | | | | ABS(ENERGY)_on_filter_BP_on_ACC_V-2 <= 0.235962: Stationary (20.0)
50 | | | | ABS(ENERGY)_on_filter_BP_on_ACC_V-2 > 0.235962: Biking (7.0/3.0)
51 | | | | ABS(VAR)_on_ACC_V-2 > 0.018555: Biking (11.0/1.0)
52 | | | | ABS(ZeroCross)_on_filter_IIR2_on_ACC_V-2 > 5
53 | | | | ABS(ENERGY)_on_filter_IIR2_on_ACC_V-2 <= 77.5625
54 | | | | ABS(MEAN)_on_ACC_V-2 <= 0.998047: Stationary (6.0)
55 | | | | ABS(MEAN)_on_ACC_V-2 > 0.998047: Biking (10.0/1.0)
56 | | | | ABS(ENERGY)_on_filter_IIR2_on_ACC_V-2 > 77.5625: Stationary (132.0/23.0)
57 ABS(ENERGY)_on_filter_IIR2_on_ACC_V-2 > 90.625
58 | | ABS(PeakToPeak)_on_ACC_V <= 0.329102
59 | | | ABS(ENERGY)_on_filter_IIR2_on_ACC_V-2 <= 92.75: Biking (25.0)
60 | | | ABS(ENERGY)_on_filter_IIR2_on_ACC_V-2 > 92.75
61 | | | ABS(ENERGY)_on_filter_BP_on_ACC_V-2 <= 0.290527
62 | | | | ABS(PeakToPeak)_on_ACC_V <= 0.319824: Stationary (18.0)
63 | | | | ABS(PeakToPeak)_on_ACC_V > 0.319824: Biking (5.0)
64 | | | | ABS(ENERGY)_on_filter_BP_on_ACC_V-2 > 0.290527: Biking (20.0/2.0)
65 | | | | ABS(PeakToPeak)_on_ACC_V > 0.329102: Stationary (12.0)
66 | | ABS(VAR)_on_ACC_V-2 > 0.021973
67 | | ABS(PeakToPeak)_on_ACC_V <= 0.352539: Biking (54.0/6.0)
68 | | ABS(PeakToPeak)_on_ACC_V > 0.352539
69 | | | ABS(MEAN)_on_ACC_V-2 <= 1.06348
70 | | | | ABS(ENERGY)_on_ACC_V-2 <= 64.9375
71 | | | | ABS(ENERGY)_on_filter_IIR2_on_ACC_V-2 <= 84.125
72 | | | | ABS(MEAN)_on_ACC_V-2 <= 1.02539
73 | | | | | ABS(ENERGY)_on_filter_BP_on_ACC_V-2 <= 0.184937: Stationary (7.0)

```

Normal text file

length:19,828 lines:256

Ln:1 Col:1 Sel:0|0

Windows (CR LF) UTF-8

INS

图 40. Weka 交叉验证

```

==== Stratified cross-validation ====
==== Summary ====

    Correctly Classified Instances      30331           96.7311 %
    Incorrectly Classified Instances   1025            3.2689 %
    Kappa statistic                   0.9421
    Mean absolute error              0.0296
    Root mean squared error          0.1202
    Relative absolute error          10.4519 %
    Root relative squared error     31.9379 %
    Total Number of Instances        31356

==== Detailed Accuracy By Class ====


|               | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC   | ROC Area | PRC Area | Class      |
|---------------|---------|---------|-----------|--------|-----------|-------|----------|----------|------------|
|               | 0.975   | 0.011   | 0.976     | 0.975  | 0.976     | 0.965 | 0.997    | 0.992    | Stationary |
|               | 0.989   | 0.028   | 0.979     | 0.989  | 0.984     | 0.962 | 0.990    | 0.986    | Walking    |
|               | 0.966   | 0.000   | 0.995     | 0.966  | 0.980     | 0.980 | 0.989    | 0.968    | Jogging    |
|               | 0.769   | 0.014   | 0.815     | 0.769  | 0.791     | 0.775 | 0.944    | 0.763    | Biking     |
| Weighted Avg. | 0.967   | 0.021   | 0.967     | 0.967  | 0.967     | 0.950 | 0.989    | 0.971    |            |



==== Confusion Matrix ====


| a    | b     | c    | d    | <-- classified as |
|------|-------|------|------|-------------------|
| 9738 | 33    | 0    | 212  | a = Stationary    |
| 12   | 17665 | 6    | 187  | b = Walking       |
| 0    | 36    | 1151 | 4    | c = Jogging       |
| 225  | 310   | 0    | 1777 | d = Biking        |


```

3.5

健身活动识别

健身活动识别旨在作为可穿戴设备（如智能手表或腕带）的健身示例。为通过决策树实现该算法，已使用安装在腕带上的设备(**LSM6DSOX**)获取了所有数据日志。

使用了两个传感器的输入（数据速率为 104 Hz 的加速度计和陀螺仪），并在具有 208 个样本的窗口中计算了两种不同的特征（均值、方差、峰到峰、最小值、最大值、过零信号），如表 11 中所示。

图 41 中由 Weka 生成的决策树可用于识别五种不同的健身活动，包括屈臂、开合跳、侧平举、俯卧撑和深蹲。

本示例中所述的健身活动识别配置以 104 Hz 运行，并在具有 208 个样本的窗口中计算特征。**LSM6DSOX** 在 1.8 V 下的电流消耗约为 563 μ A。关闭机器学习内核时，**LSM6DSOX**（采用 104 Hz 的加速度计和陀螺仪）的电流消耗约为 550 μ A，因此该算法的机器学习内核的额外电流消耗为 13 μ A。

表 11. 健身活动识别配置

配置	加速度计, 104 Hz ODR, 4 g 满量程
	陀螺仪, 104 Hz ODR, 2000 dps 满量程
窗口长度	208 个样本（约 2 秒）
特征	均值
	方差
	峰到峰
	最小值
	最大值
	过零信号
输出	无活动 (0)
	屈臂 (4)
	开合跳 (5)
	侧平举 (6)
	俯卧撑 (7)
	深蹲 (8)
元分类器	无活动时为 0
	所有其他输出为 2

图 41. 健身活动识别决策树

```

gym_activity_recognition.txt - Notepad
File Edit Format View Help

MEAN_on_ACC_Z <= 0.155976
| MEAN_on_ACC_Z <= 0.0731812
|   ABS(PeakToPeak)_on_ACC_Z <= 0.561523: no_activity
|   ABS(PeakToPeak)_on_ACC_Z > 0.561523
|     ABS(PeakToPeak)_on_ACC_X <= 0.98291: no_activity
|     ABS(PeakToPeak)_on_ACC_X > 0.98291
|       ABS(PeakToPeak)_on_GY_Y <= 3.07227: no_activity
|       ABS(PeakToPeak)_on_GY_Y > 3.07227
|         ABS(PeakToPeak)_on_GY_Y <= 12.0
|           ABS(PeakToPeak)_on_GY_X <= 7.55469
|             ABS(PeakToPeak)_on_GY_Z <= 6.21094: bicep_curls
|             ABS(PeakToPeak)_on_GY_Z > 6.21094: no_activity
|           ABS(PeakToPeak)_on_GY_X > 7.55469: no_activity
|         ABS(PeakToPeak)_on_GY_Y > 12.0: no_activity
MEAN_on_ACC_Z > 0.0731812: no_activity

MEAN_on_ACC_Z > 0.155976
MEAN_on_ACC_Z <= 0.864257
| ABS(PeakToPeak)_on_ACC_Z <= 0.465088: no_activity
| ABS(PeakToPeak)_on_ACC_Z > 0.465088
|   ABS(PeakToPeak)_on_ACC_X <= 0.696289: no_activity
|   ABS(PeakToPeak)_on_ACC_X > 0.696289
|     ABS(PeakToPeak)_on_GY_Y <= 2.07617: no_activity
|     ABS(PeakToPeak)_on_GY_Y > 2.07617
|       ABS(PeakToPeak)_on_GY_Y <= 8.0
|         ABS(PeakToPeak)_on_GY_X <= 7.13281
|           ABS(PeakToPeak)_on_GY_Z <= 3.05273: lateral_raises
|           ABS(PeakToPeak)_on_GY_Z > 3.05273: no_activity
|         ABS(PeakToPeak)_on_GY_X > 7.13281: no_activity
|       ABS(PeakToPeak)_on_GY_Y > 8.0: no_activity
MEAN_on_ACC_Z > 0.864257
| ABS(PeakToPeak)_on_ACC_Z <= 0.461426: no_activity
| ABS(PeakToPeak)_on_ACC_Z > 0.461426
|   ABS(PeakToPeak)_on_ACC_Z <= 2.4
|     ABS(PeakToPeak)_on_ACC_Y <= 1.2207
|       ABS(PeakToPeak)_on_ACC_X <= 1.03809: squats

```

3.6

示例汇总

下表汇总显示了本文档中所述的所有示例，以显示 LSM6DSOX 中的机器学习内核在不同配置下的典型电流消耗。主要贡献者为机器学习内核 ODR（可能与传感器 ODR 不同）、配置的决策树数量以及每个决策树的节点数。

表 12. 示例汇总

算法	MLC_ODR	决策树 数量	节点数	MLC 额外电流消耗
振动监测	26 Hz	1	2	1 μ A
运动强度	12.5 Hz	1	7	1 μ A
6D 位置识别	26 Hz	1	8	2 μ A
智能手机应用程序的活动识别	26 Hz	1	126	4 μ A
健身活动识别	104 Hz	1	19	13 μ A

附录 A Weka

Weka 是由新西兰的怀卡托大学开发的免费软件。它包含一系列用于数据分析和预测建模的可视化工具与算法，以及用于轻松访问这些功能的图形用户界面。

Weka 是最受欢迎的决策树生成机器学习工具之一。本节包含有关该外部软件的一些详细信息，可在以下链接中找到其他详细信息：

- [Weka 下载](#)
- [Weka 网站](#)
- [Weka 用户指南](#)

所有 Weka 技术均基于以下假设，即数据可作为一个平面文件或关系来使用，并通过固定数量的属性来描述每个数据点。

ARFF（属性关系文件格式）文件是 ASCII 文本文件，描述了共享一组属性的实例列表。ARFF 文件具有两个不同的部分，如图 42 中所示：包含属性（特征、类别）的头部分，包含所有特征值以及与该特征集相关的相应类别的数据部分。

图 42. ARFF 示例

The screenshot shows a Windows Notepad++ window displaying an ARFF file. The file starts with a header section defining a relation and its attributes:

```
1 #relation "MLQ"
2
3 #attribute ABS(WEAN)_on_ACC_X numeric
4 #attribute ABS(WEAN)_on_ACC_Y numeric
5 #attribute ABS(WEAN)_on_ACC_Z numeric
6 #attribute ABS(VAR)_on_ACC_X numeric
7 #attribute ABS(VAR)_on_ACC_Y numeric
8 #attribute ABS(VAR)_on_ACC_Z numeric
9 #attribute ABS(VAR)_on_ACC_V numeric
10 #attribute ABS(VAR)_on_ACC_V2 numeric
11 #attribute ABS((EnergyV)_on_filter_BP_on_ACC_V) numeric
12 #attribute ABS((EnergyV)_on_filter_BP_on_ACC_V2) numeric
13 #attribute ABS((EnergyV)_on_filter_LP_on_ACC_V) numeric
14 #attribute ABS((ZeroCross)_on_filter_BP_on_ACC_V) numeric
15 #attribute ABS((ZeroCross)_on_filter_LP_on_ACC_V) numeric
16 #attribute ABS(PeakToPeak)_on_ACC_V numeric
17 #attribute class {Stationary, Walking, Jogging, Biking}
```

Following the header is a data section labeled '#data' containing approximately 45 data instances, each with 17 numerical values and a class label at the end.

图 43. Weka GUI Chooser



启动 Weka 时，将显示 Weka GUI Chooser 窗口(图 43)，可通过第一个按钮选择的“Explorer”部分是 Weka 的主用户界面。

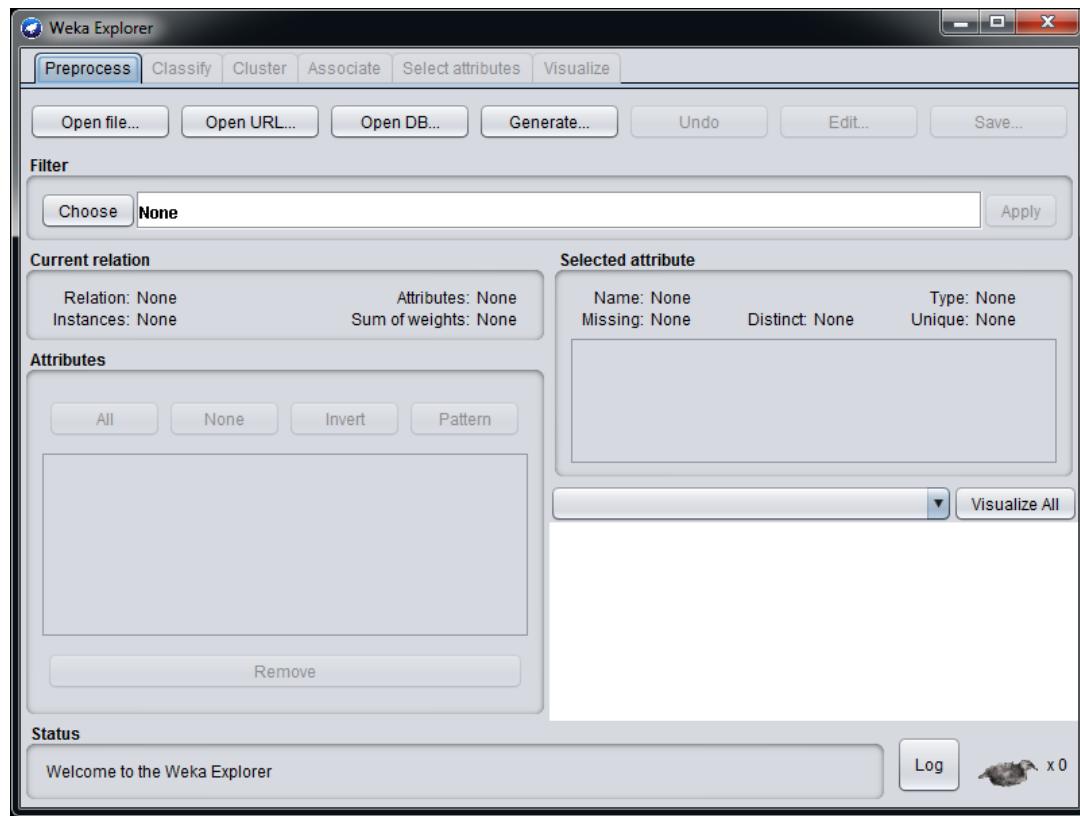
选择 Weka Explorer，将出现一个新界面(图 44)。Explorer 界面中有几个面板：

- 预处理面板具有导入数据的工具。
- 分类面板允许将分类和回归算法应用于数据集，以估计所产生的预测模型的精度并显示错误预测。
- 通过群集面板可访问 Weka 中的群集技术。
- 通过关联面板可访问关联规则学习者，这些学习者试图识别数据中属性之间的所有重要的相互关系。
- 选择属性面板提供了用于识别数据集中最具预测性的属性的算法。
- 可视化面板显示了散点图矩阵。

该附录部分仅描述了预处理和分类面板。

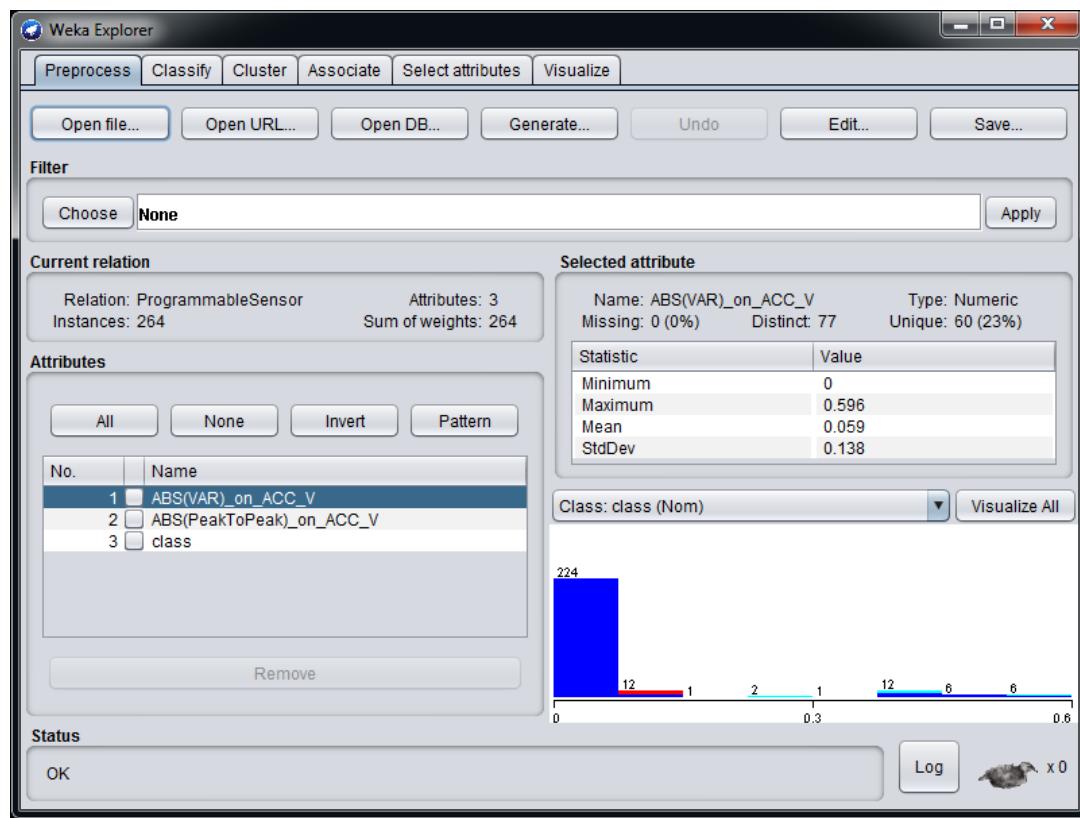
预处理面板如图 44 所示，它允许通过“Open file”按钮加载 ARFF 文件。

图 44. Weka Explorer



加载 ARFF 文件后，“预处理”面板将显示导入的 ARFF 文件的所有属性（功能和类别）。属性可以图形化方式显示，用户可以选择用于分类的属性。

图 45. Weka Explorer - 属性



选择属性后，可在 Weka Explorer (图 46) 的分类面板中配置分类器。Weka 中有许多可用的分类器：通过选择分类器 J48 (在“树”下)，可以生成决策树(图 47)。

图 46. Weka 分类

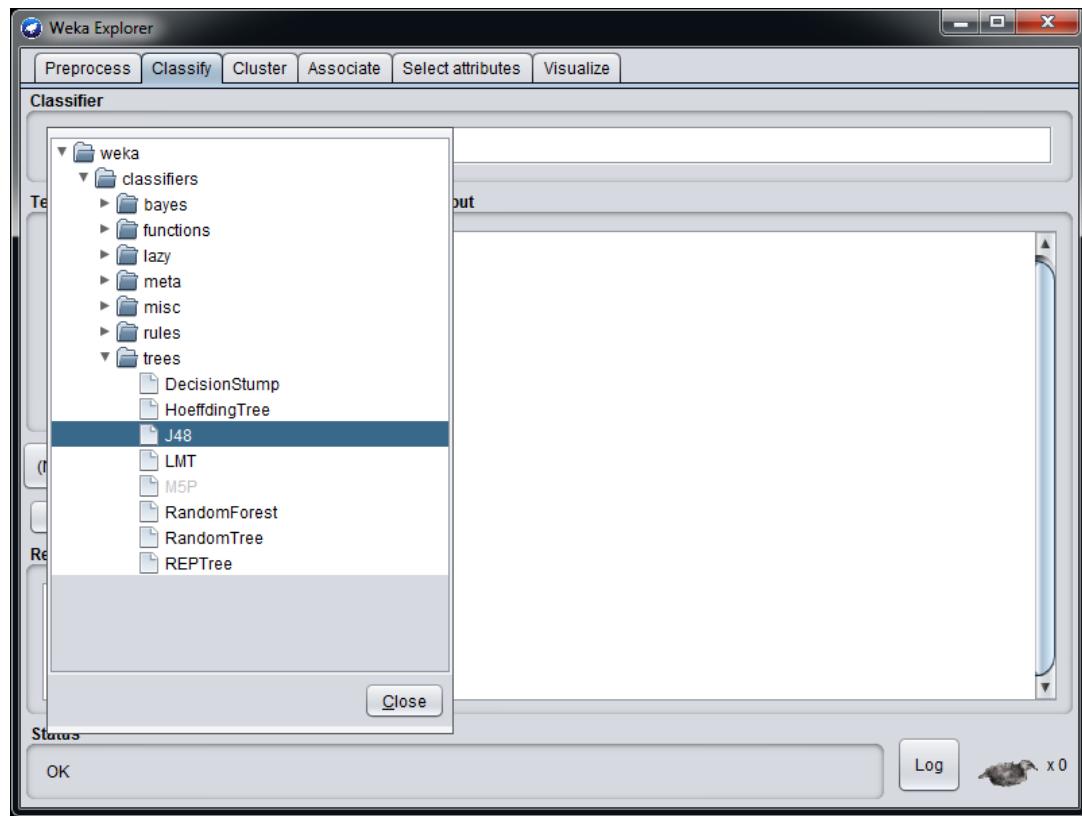
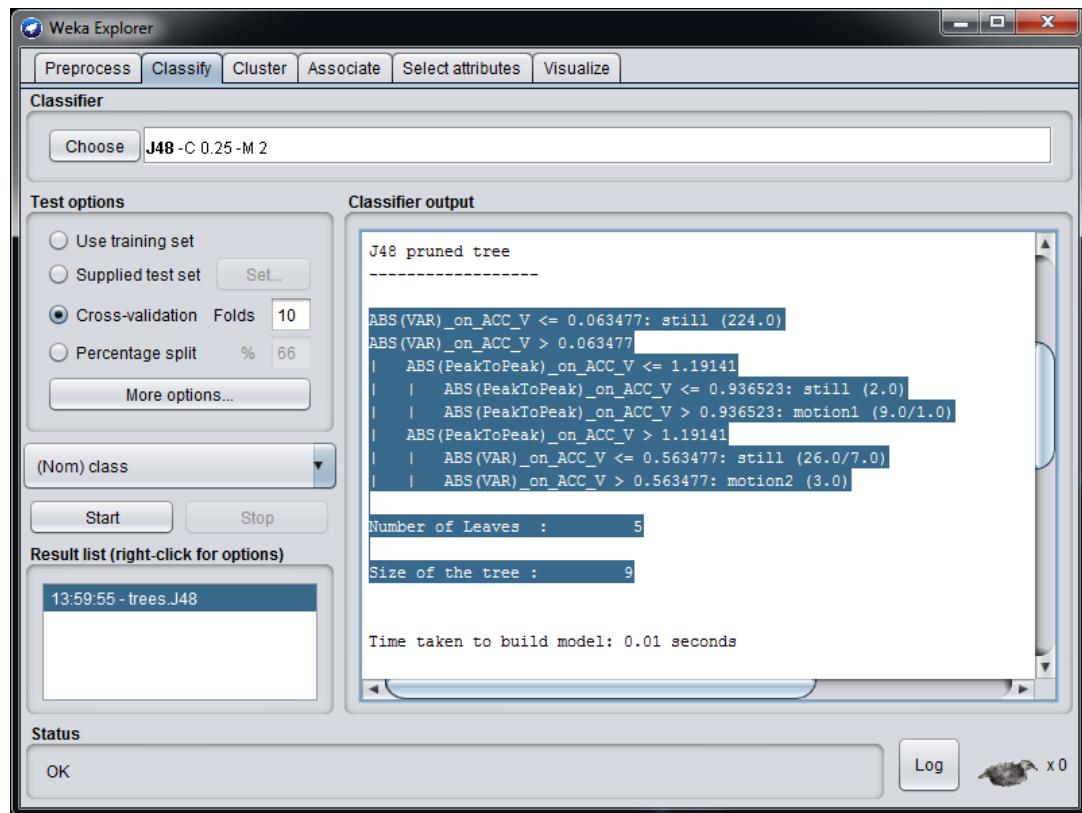
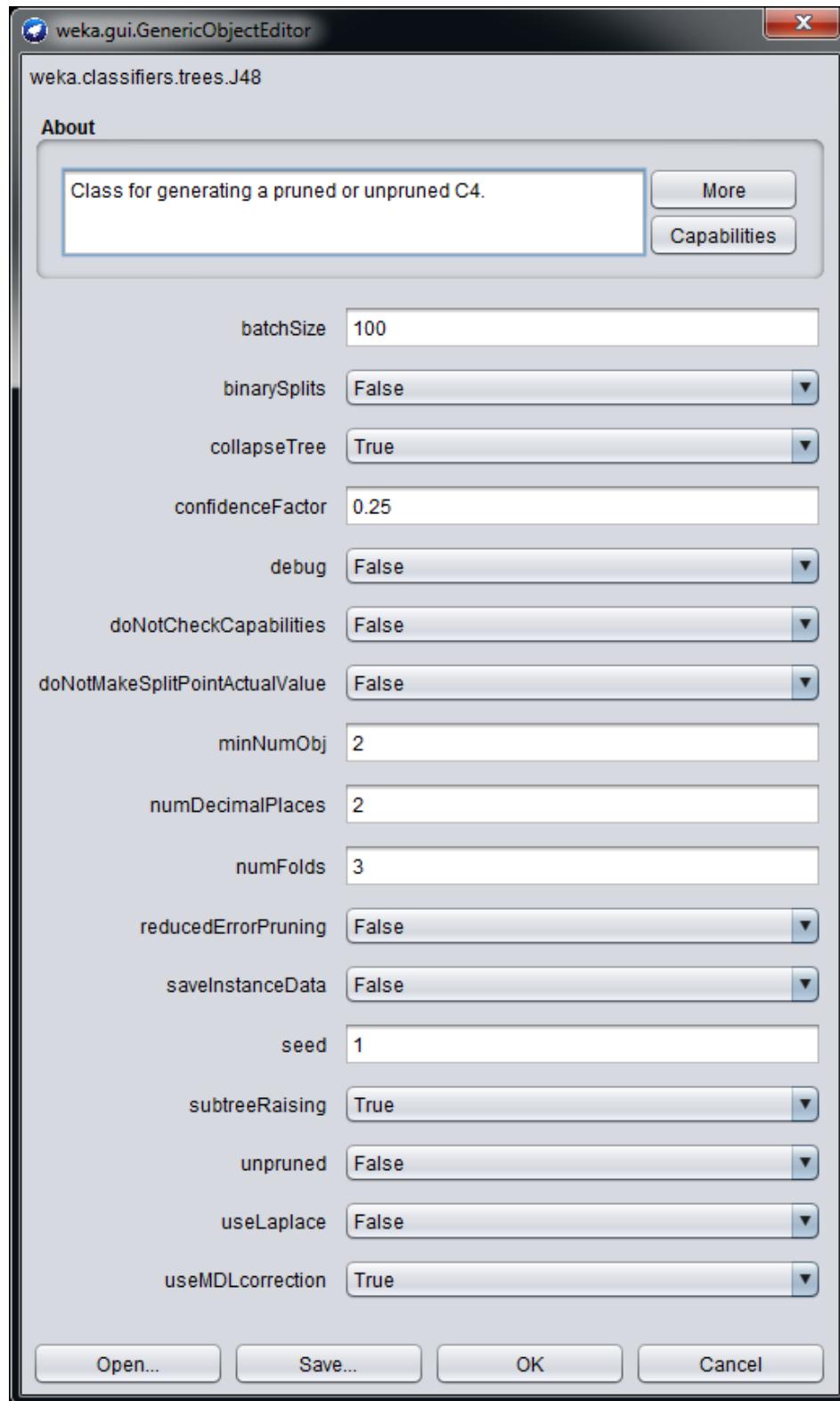


图 47. Weka 分类 J48



可更改分类器部分中的许多参数(图 48)，通过点击“Start”按钮，可生成不同的决策树（参见图 47）。

图 48. Weka J48 分类器参数



可轻松比较所有生成的决策树的以下方面：

- 节点数

由于 Weka 中的 J48 算法生成的决策树为二叉树，因此可通过从决策树之后的第一行出现的“叶数”参数减 1 来获得节点数（参见图 49. 正确分类实例）。

- 正确分类实例

这是对所创建模型的精度的估计。将模型的结果与标签提供的结果进行比较。图 49. 正确分类实例 显示了活动识别模型的正确分类实例。

- 混淆矩阵

NxN 表总结了分类模型预测的成功程度，即标签与模型分类的相关性。混淆矩阵的一个轴为预测模型标签，另一个轴为实际标签。

图 49. 正确分类实例

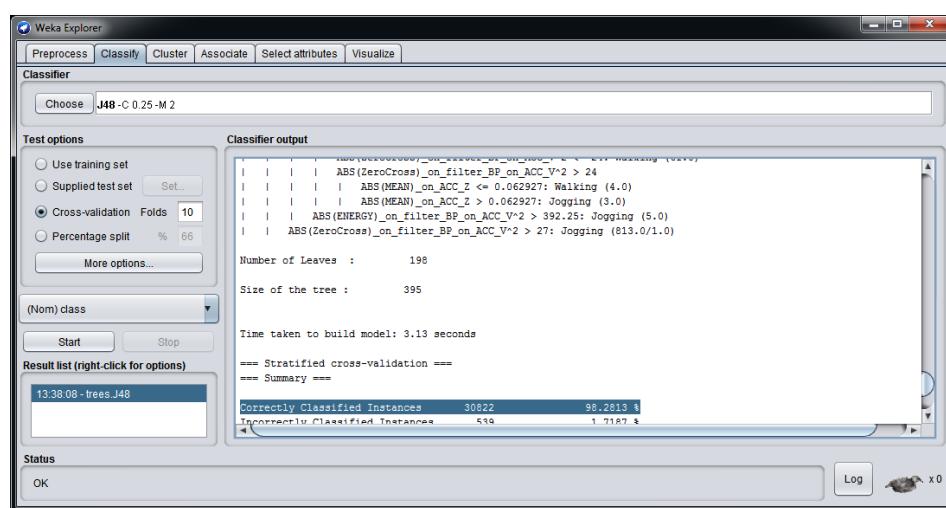
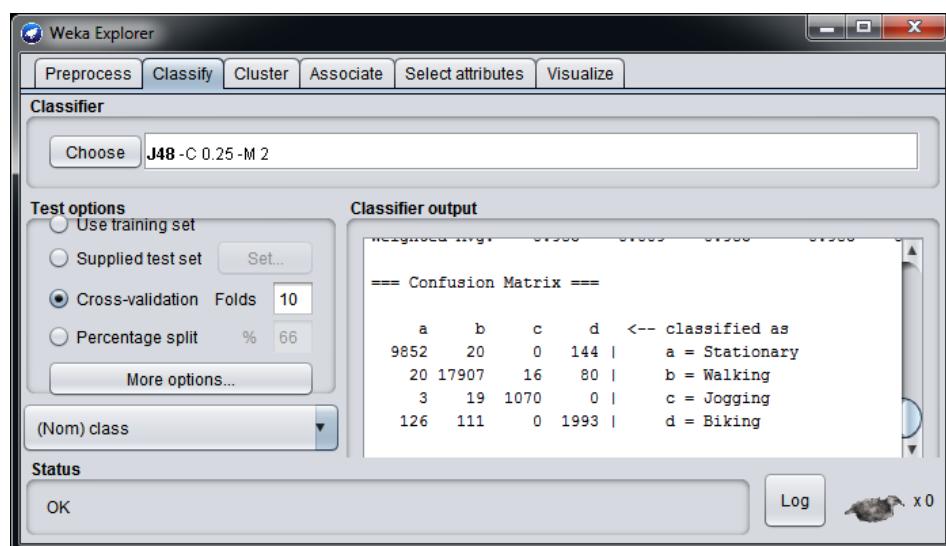


图 50. 混淆矩阵 显示了具有四个类别（静止、行走、慢跑、骑自行车）的活动识别算法的混淆矩阵示例。

图 50. 混淆矩阵



附录 B RapidMiner

RapidMiner 是一种数据科学软件平台，可为数据准备、机器学习、深度学习、文本挖掘以及预测分析提供集成环境。它用于商务应用以及验证、教育、培训、快速原型和应用开发，并支持机器学习过程的所有阶段，包括数据准备、结果可视化、模型验证和优化。

本附录描述了使用 RapidMiner Studio 从 ARFF 文件生成决策树的过程。为此，考虑了洗手检测算法的简单示例。打开 RapidMiner Studio 之后，主要步骤如下：

1. 从左侧的“Operators”窗口添加“Open File”运算符，并将运算符拖到空白“过程”窗口中，如图 51 中所示。
2. 双击“Open File”运算符，以选择要载入的 ARFF 文件。
3. 找到“Read ARFF”运算符，并将其拖到“Process”窗口中。然后将“Read ARFF”运算符连接至“Open File”运算符，如图 52 中所示。
4. 找到“Set Role”运算符并将其拖动到“Process”窗口中。然后，双击“Set Role”运算符，并在“Parameters”窗口中键入属性名称和目标角色，如图 53 中所示。
5. 找到“Decision Tree”运算符并设置相应的参数，如图 54 中所示。您还需要将“Decision Tree”运算符连接到“res”。
6. 单击 RapidMiner Studio 左上方的“Run”按钮（蓝色三角形图标）。
7. 点击“Run”按钮后，“Results”选项卡将根据“Graph”（图 55）和“Description”显示生成的决策树。
8. 在生成的决策树的“Description”部分中（图 56），您需要将决策树复制到文本文件中，可将该文件导入到 Unico 中的 MLC 工具中。

图 51. RapidMiner Studio - 打开文件

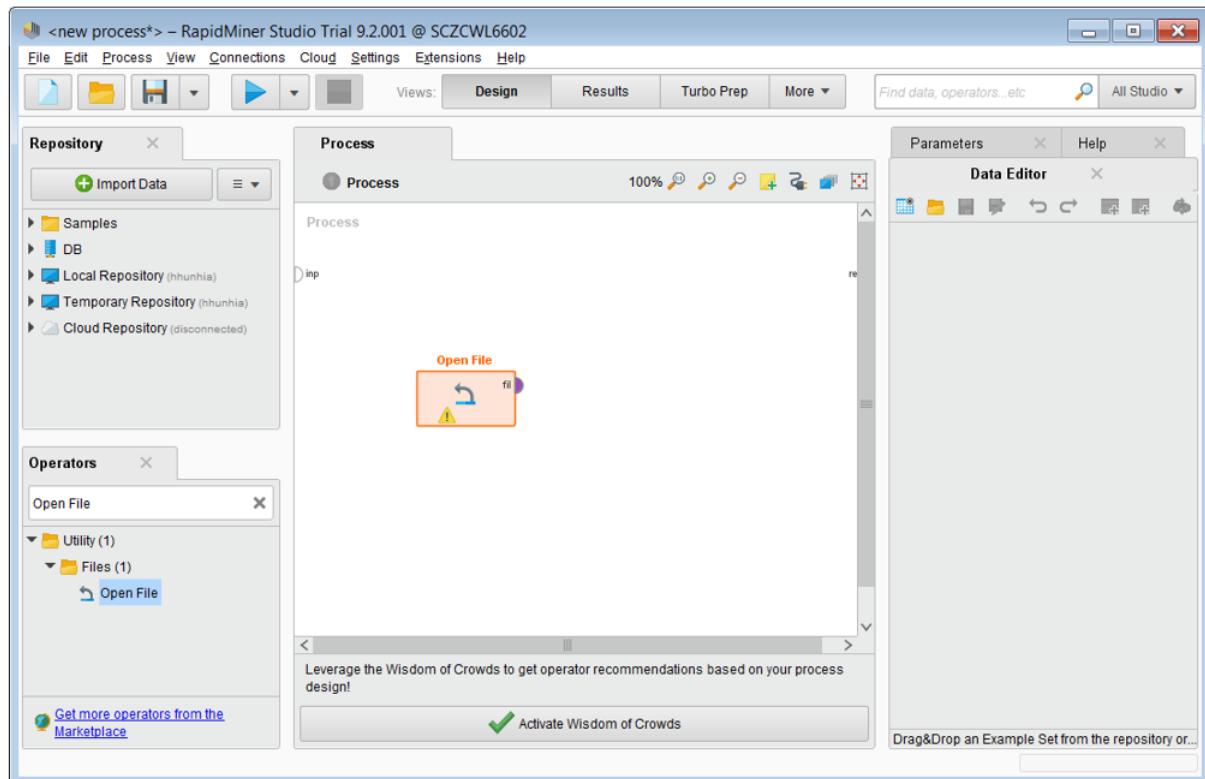


图 52. RapidMiner Studio - 读取 ARFF

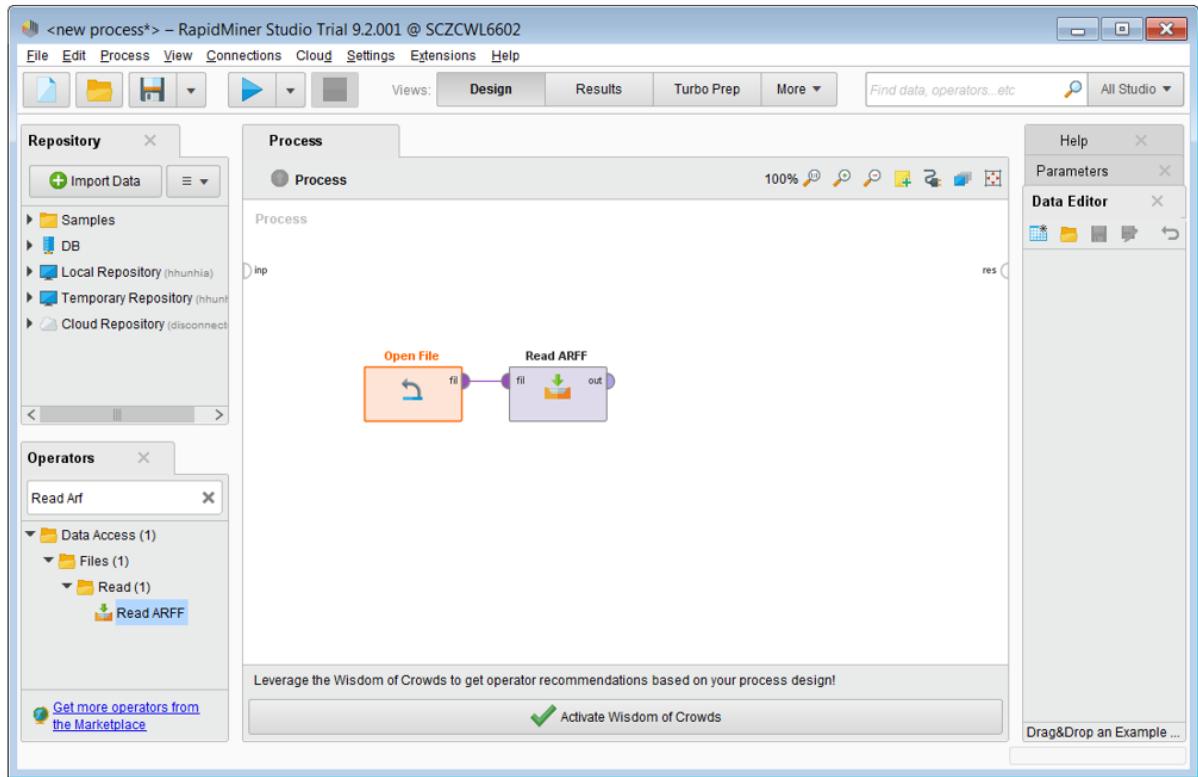


图 53. RapidMiner Studio - 设置角色

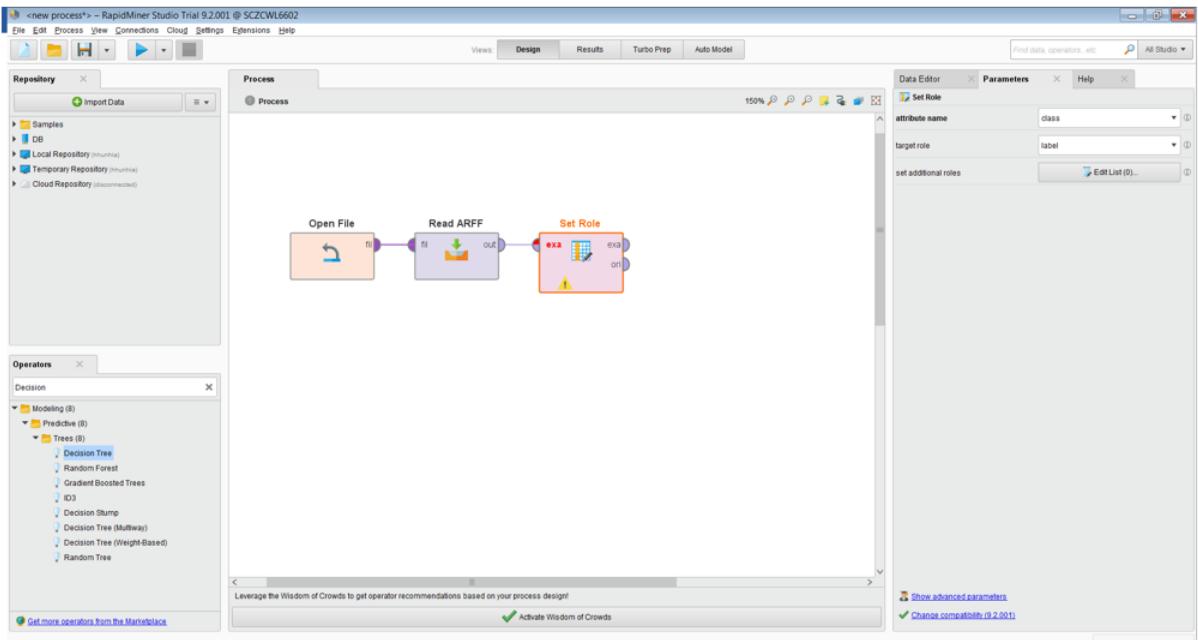


图 54. RapidMiner Studio - 决策树运算符

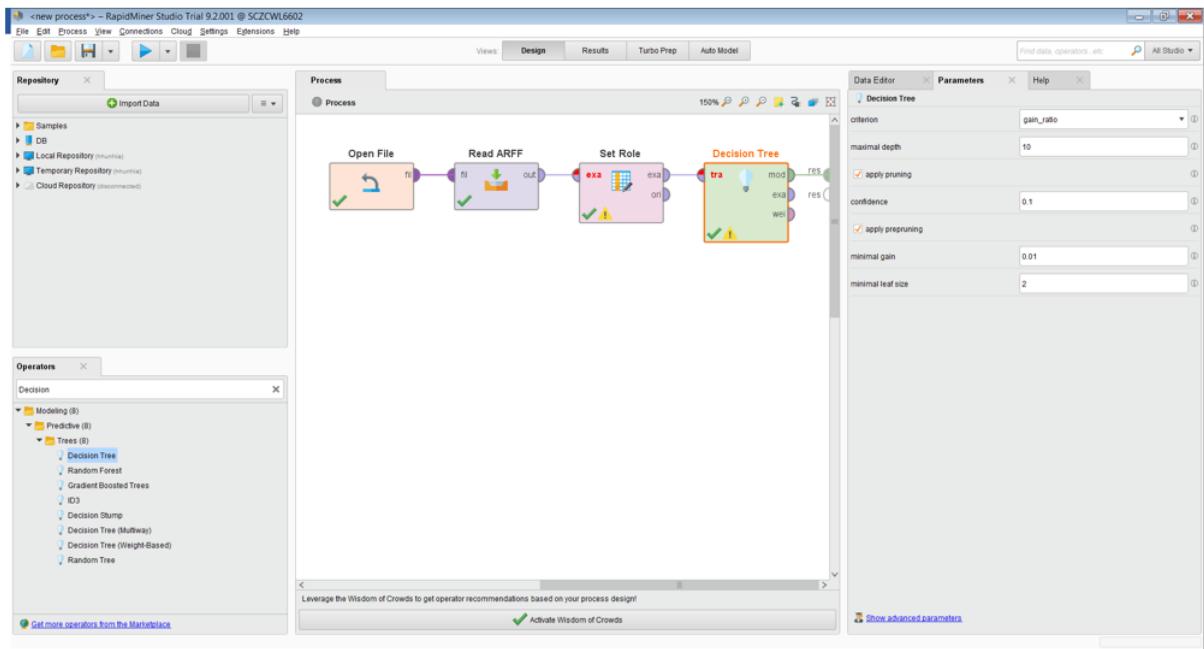


图 55. RapidMiner Studio - 决策树图

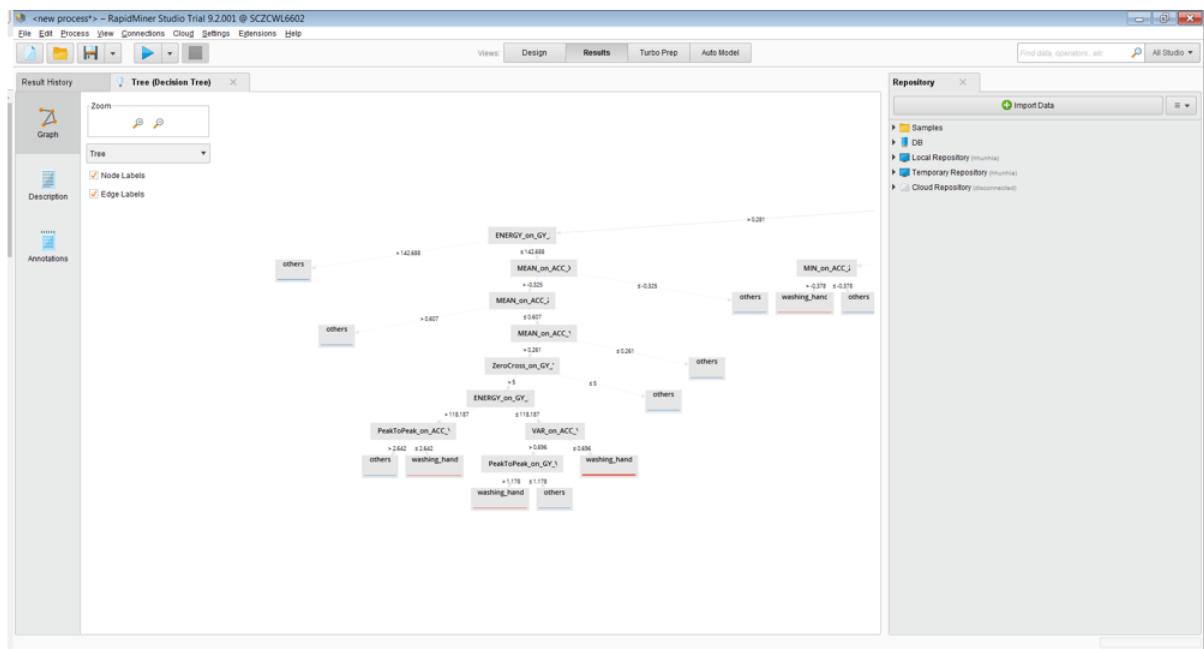
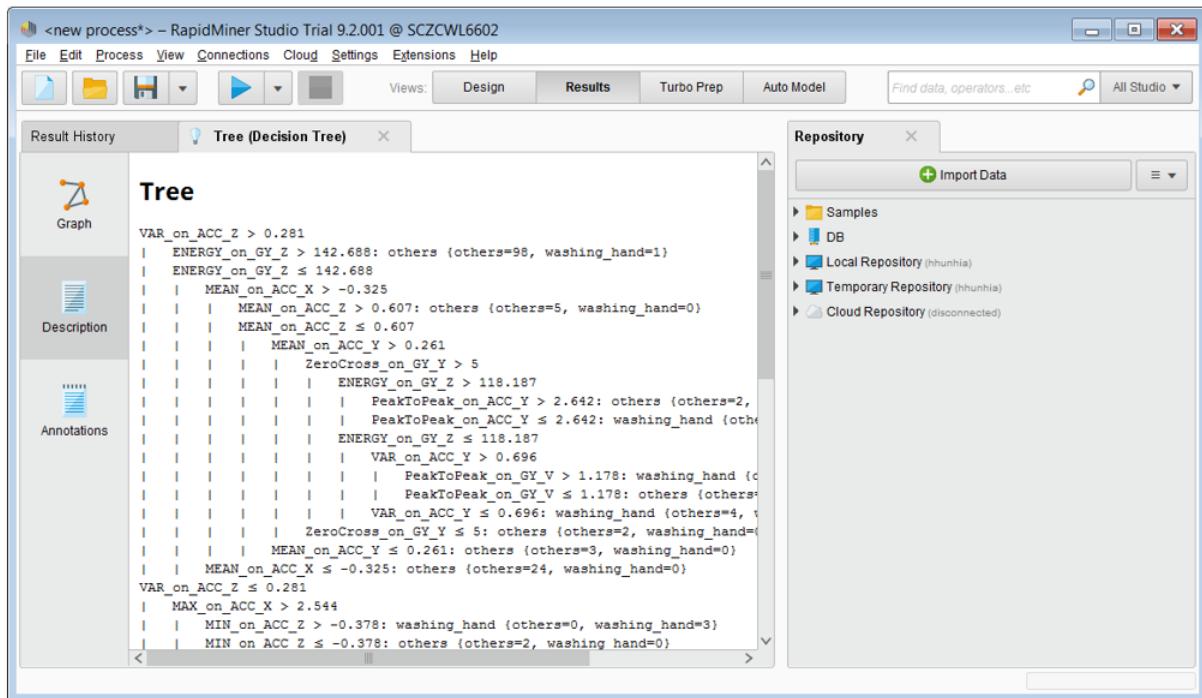


图 56. RapidMiner Studio - 决策树说明



附录 C Matlab

可使用 Matlab 生成机器学习内核的决策树。Matlab 上提供 Matlab 专用脚本。

将全部脚本导入 Matlab 工作区后，调用“*Generate_DcisionTree()*”函数，并指定两个文件名（一个为.arff 文件，其中包含通过 Unico 中的机器学习内核工具计算出的特征，另一个为.txt 文件，其中包含生成的决策树）：

```
filename_ARFF = 'features.arff';
filename_dectree = 'decision_tree.txt';
Generate_DcisionTree(filename_ARFF, filename_dectree);
```

可在 GitHub 存储库的“matlab”文件夹所包含的 *README.md* 文件中找到更多详细信息。

附录 D Python

可通过“*scikit*”包使用 Python 生成机器学习内核的决策树。

Python 即可提供作为 Jupyter 记事本(*.ipynb)的 Python 脚本，也可以提供通用 Python 脚本(*.py)。可在 GitHub 存储库的“*python*”文件夹所包含的 *README.md* 文件中找到更多详细信息。

附录 E 词汇表

本节包含用于机器学习的词汇表。大多数词汇表取自 <https://developers.google.com/machine-learning/glossary/>。

ARFF	ARFF （属性关系文件格式）文件是 ASCII 文本文件，描述了共享一组属性的实例列表。 ARFF 文件由怀卡托大学计算机科学系的机器学习项目部开发，用于 Weka 机器学习软件。
属性/特征	属性为实例的一个方面（如温度、湿度）。在机器学习中，属性通常被称为特征。特殊属性为定义该示例所属类别的类别标签（在监督学习中是必须的）。
二进制分类	输出两个互斥类别之一的一种分类任务。例如，评估电子邮件消息，并输出“垃圾邮件”或“非垃圾邮件”的机器学习模型为二进制分类器。
分类	标签的一组枚举目标值之一。例如，在检测垃圾邮件的二进制分类模型中，两种类别为垃圾邮件和非垃圾邮件。在识别狗的品种的多类别分类模型中，类别为贵宾犬、小猎犬、哈巴狗等。
分类模型	用于区分两个或多个离散类别的机器学习模型类型。例如，自然语言处理分类模型可确定输入的句子为法语、西班牙语还是意大利语。
分类阈值	一种标量值标准，该标准应用于模型预测分数，以将阳性类别和阴性类别分离。在将逻辑回归结果映射到二进制分类时使用。例如，考虑一个用于确定给定电子邮件消息是否为垃圾邮件的逻辑回归模型。如果分类阈值为 0.9，则逻辑回归值高于 0.9 的邮件被视为垃圾邮件，逻辑回归值低于 0.9 的邮件被视为非垃圾邮件。
类别不平衡数据集	两个类别的标签具有明显不同的频率的二进制分类问题。例如，0.0001 个示例具有阳性标记且 0.9999 个示例具有阴性标记的疾病数据集为类别不平衡问题，但 0.51 个示例标记一个球队获胜而 0.49 个示例标记另一个球队获胜的足球比赛预测器为类别不平衡问题。
限幅	一种用于处理异常值的技术。具体而言，将大于最大设置值的特征值减小为该最大值。同样，将小于特定最小值的特征值增加至该最小值。
混淆矩阵	$N \times N$ 表总结了分类模型预测的成功程度，即标签与模型分类的相关性。混淆矩阵的一个轴为预测模型标签，另一个轴为实际标签。
交叉验证	一种通过根据训练集中保留的一个或多个非重叠数据子集测试模型来估计模型泛化为新数据的程度的机制。
数据分析	通过考虑样本、测试与可视化来了解数据。在建立第一个模型之前首次接收到数据集时，数据分析可能非常有用。这对于了解实验和调试系统问题也至关重要。
数据扩增	通过转换现有示例来创建其他示例，可人为扩增培训示例的范围和数量。例如，假设图像是您的特征之一，但您的数据集没有供模型学习有用的关联的足够图像示例。理想情况下，您将为数据集添加足够的标签图像，以正确训练模型。如果这不可行，数据扩增可通过旋转、拉伸和反射每个图像来产生原始图片的多个变体，从而可能产生足够的带标签数据，以便实现出色的训练。
数据组或数据集	示例集。
决策边界	模型在二元类别或多类别分类问题中学到的类别之间的分隔符。
决策阈值	分类阈值的同义词。
决策树	以分支语句序列表示的模型。
离散特征	具有一组有限可能值的特征。例如，其值只能为动物、植物或矿物的特征为离散（或类属）特征。与连续特征形成对比。
判别模型	通过一组一个或多个特征预测标签的模型。更正式地讲，判别模型在给定特征和权重下定义了输出条件概率。
下采样	可能表示以下任一情况的重载术语： <ul style="list-style-type: none">• 减少特征中的信息量，以更有效地训练模型。• 以不成比例的低百分比训练超代表性类别示例，以改善代表性不足的类别的模型训练。
动态模型	以持续更新方式在线训练的模型。即，将数据连续输入模型。
示例	一行数据集。包含一个或多个特征，也可能包含一个标签的示例。另请参见带标签示例和不带标签的示例。
假阴性 (FN)	模型错误地预测阴性类别的示例。例如，该模型推断出特定电子邮件不是垃圾邮件（阴性类别），但该电子邮件实际上为垃圾邮件。
假阳性 (FP)	模型错误地预测阳性类别的示例。例如，该模型推断出特定电子邮件为垃圾邮件（阳性类别），但该电子邮件实际上不是垃圾邮件。

假阳性率 (FPR)	ROC 曲线中的 x 轴。假阳性率定义如下： $\text{假阳性率} = \text{假阳性} / (\text{假阳性} + \text{真阴性})$
特征	用于进行预测的输入变量。
特征工程	确定那一特征可能对训练模型有用并将日志文件和其他来源的原始数据转换为所述特征的过程。 特征工程有时也称为特征提取。
特征提取	具有以下定义之一的重载术语： <ul style="list-style-type: none">• 检索通过无监督或预先训练的模型计算出的中间特征表示，以用作另一个模型的输入。• 特征工程的同义词。
特征集	机器学习模型训练的一组特征。例如，邮政编码、财产规模和财产状况可能包括一个用于预测房价的模型的简单特征集。
泛化	是指模型对以前看不到的新数据（与用于训练模型的数据完全不同）做出正确预测的能力。
真相	正确答案。现实。由于现实通常是主观的，因此评估专家通常为真相的代理。
启发法	问题的快速解决方案，可能是最佳解决方案，也可能不是最佳解决方案。
不平衡数据集	类别不平衡数据集的同义词。
独立且恒等分布 (i.i.d.)	从恒定分布中提取的数据，每个绘制的值都不依赖于先前绘制的值。 i.i.d. 是一种理想的机器学习条件——是一种有用的数学构造，但在现实世界中几乎找不到。例如，网页的访客分布可能为 i.i.d. 在短暂的时间内，即在那个短暂的窗口内分布不会改变，一个人的访问通常独立于另一个人的访问。但是，如果扩大了该时间窗口，则网页访问者的季节性差异可能会出现。
推论	在机器学习中，通常是指通过将训练后的模型应用于不带标签的示例来进行预测的过程。在统计学上，推论是指根据某些观察到的数据拟合分布参数的过程。
实例	例子的同义词。
可解释性	模型预测的解释难易程度。深度模型通常无法解释；也就是说，深层模型的不同层可能难以解读。相比之下，线性回归模型和广义模型通常更具可解释性。
J48	C4.5 算法的开源 Java 实现
标签	在监督式学习中，示例的“答案”或“结果”部分。带标签数据集中的每个示例均包含一个或多个特征及一个标签。例如，在房屋数据集中，特征可能包括卧室数量、浴室数量、房屋使用年限，而标签可能为房屋价格。在垃圾邮件检测数据集中，特征可能包括主题行、发件人和电子邮件本身，而标签可能为“垃圾邮件”或“非垃圾邮件”。
线性回归	通过输入特征的线性组合输出连续值的一种回归模型。
机器学习	通过输入数据构建（训练）预测模型的程序或系统。通过从训练模型时的相同分布获得的新（从未见过）数据，该系统使用学得的模型来做出有用的预测。机器学习还指与这些程序或系统有关的研究领域。
多数类别	类别不平衡数据集中的更常见标签。例如，假定数据集包含 99% 的非垃圾邮件标签和 1% 的垃圾邮件标签，则非垃圾邮件标签为多数标签。
Matplotlib	一种开源的 Python 2D 绘图库。 matplotlib 可帮助您可视化机器学习的不同方面。
少数类别	类别不平衡数据集中的较不常见标签。例如，假定数据集包含 99% 的非垃圾邮件标签和 1% 的垃圾邮件标签，则垃圾邮件标签为少数标签。
ML	机器学习的缩写。
模型训练	确定最佳模型的过程。
多类别分类	区分两个以上的类别的分类问题。例如，大约有 128 种枫树树种，因此对枫树树种进行分类的模型为多类别模型。相反，将电子邮件仅分为两类（垃圾邮件和非垃圾邮件）的模型为二进制分类模型。
多项式分类	多类分类的同义词。
阴性类别	在二进制分类中，一种类别被称为阳性类别，另一种类型被称为阴性类别。阳性类别为我们寻找的类别，而阴性类别则为另一种可能。例如，医学测试中的阴性类别可能为“非肿瘤”。电子邮件分类器中的阴性类别可能为“非垃圾邮件”。另请参阅阳性类别。
神经网络	一种从大脑获得灵感的模型，该模型分为多层（至少有一层是隐藏的），这些层包含简单的连接单元或神经元以及非线性元素。
节点（决策树）	对属性的“测试”。
噪声	广义上讲，任何会使数据集中的信号模糊的东西。噪声可通过多种方式引入数据中。例如：

	<ul style="list-style-type: none"> 评分员在标签上犯错误。 人员和仪器错误记录或忽略了特征值。
归一化	将实际值范围转换为标准值范围（通常为-1 到+1 或 0 到 1）的过程。例如，假设某个特征的自然范围为 800 到 6,000。通过减法和除法，您可以将这些值归一化为-1 到+1。 另请参见缩放。
数值数据	以整数或实数值表示的特征。
异常值	<p>偏离大多数其他值的值。在机器学习中，以下任何值均为异常值：</p> <ul style="list-style-type: none"> 具有高绝对值的权重。 与实际值相差甚远的预测值。 与平均值大约相差 3 个标准偏差以上的输入数据值。 <p>异常值通常会在模型训练中引起问题。限幅为一种管理异常值的方式。</p>
过度拟合	创建的模型与训练数据的匹配度过高，以至于该模型无法对新数据做出正确预测。
参数	ML 系统自行训练的模型变量。
性能	<p>重载术语具有以下含义：</p> <ul style="list-style-type: none"> 软件工程中的传统含义。即：该软件运行速度有多快（或效率有多高）？ 在 ML 中的含义性能在这里回答了以下问题：该模型有多正确？也就是说，模型的预测效果如何？
阳性类别	<p>在二进制分类中，将两种可能的类别分别标记为阳性和阴性。阳性结果是我们要测试的东西。（诚然，我们同时测量两种结果，但要配合）例如，医学测试中的阳性类别可能为“肿瘤”。电子邮件分类器中的阳性类别可能为“垃圾邮件”。</p> <p>与阴性类别形成对比。</p>
精度	分类模型的指标。精度用于确定预测阳性类别时的模型正确频率。即： $\text{精度} = \text{真阳性} / (\text{真阳性} + \text{假阳性})$
预测	提供输入示例时的模型输出。
预训练模型	已经训练过的模型或模型组件。
代理标签	<p>用于近似表示数据机中不直接可用的标签的数据。</p> <p>例如，假设您将“会下雨吗？”作为数据集的布尔标签，但是该数据集不包含降雨数据。如果有照片，您可以将带伞的人的照片作为“会下雨吗？”的代理标签。但是，代理标签可能会使结果失真。例如，在某些地方，携带雨伞防晒比防雨淋更常见。</p>
评分员	在示例中提供标签的人。有时被称为“注解者”。
召回率	<p>分类模型的指标，可回答以下问题：</p> <p>“在所有可能的阳性标签中，模型可以正确识别多少个标签？”</p> <p>即：</p> <p>$\text{召回率} = \text{真阳性} / (\text{真阳性} + \text{假阴性})$</p>
回归模型	一种输出连续（通常为浮点）值的模型。相比之下，分类模型输出离散值。如“薰草”或“卷丹”。
强化学习	一种机器学习算法，该算法通过一系列操作后的反馈（奖励和惩罚）来实现最终奖励的最大化。例如，大多数游戏的最终奖励为胜利。通过评估最终导致获胜的先前游戏动作序列以及最终导致失败的序列，强化学习系统可成为玩游戏的专家。
表示	将数据映射到有用特征的过程。
ROC 曲线	<p>$\text{ROC} = \text{接收器的工作特征}$</p> <p>真阳性率与假阳性率在不同分类阈值下的对比曲线。</p>
缩放	<p>特征工程中的一种常用做法，即通过调整特征的值范围，以便与数据集中的其他特征的范围匹配。例如，假如您希望数据集中的所有浮点特征的范围为 0 到 1。给定的特定特征的范围为 0 到 500，您可以通过将每个值除以 500 来缩放该特征。</p> <p>另请参见归一化。</p>
Scikit-learn	流行的开源 ML 平台。参见 www.scikit-learn.org 。
打分	推荐系统的一部分，为候选生成阶段所产生的每个项目提供一个值或排名。

半监督式学习	根据数据训练模型，其中一些训练示例有标签，而其他示例无标签。一种半监督式学习技术是，为不带标签的示例推断标签，然后根据推断出的标签进行训练，以创建新模型。如果获取标签的成本很高，而不带标签的示例很多，则半监督式学习可能很有用。
序列模型	其输入具有序列依赖性的模型。例如，根据之前观看的一系列视频来预测下一个观看的视频。
服务	推断的同义词。
静态模型	离线训练的模型。
平稳性	数据集中的数据属性，其中数据分布在一个或多个维度上保持不变。最常见的是，如果维度是时间，则意味着呈现出平稳性的数据不会随时间而变化。例如，呈现出平稳性的数据从 9 月到 12 月没有变化。
监督式机器学习	根据输入数据及其相应的标签训练模型。监督式机器学习类似于学生通过研究一组问题及其相应的答案来学习一门学科。在掌握问题与答案之间的映射后，学生可以为同一主题的新问题（从未见过）提供答案。与无监督式机器学习进行比较。
目标	标签的同义词。
训练	确定组成模型的理想参数的过程。
训练集	用于训练模型的数据集的子集。 与验证集和测试集形成对比。
真阴性 (TN)	模型正确地预测阴性类别的示例。例如，该模型推断出特定电子邮件不是垃圾邮件，而该电子邮件确实不是垃圾邮件。
真阳性 (TP)	模型正确地预测阳性类别的示例。例如，该模型推断出特定电子邮件为垃圾邮件，而该电子邮件确实为垃圾邮件。
真阳性率 (TPR)	召回率的同义词。即： $\text{真阳性率} = \text{真阳性} / (\text{真阳性} + \text{假阴性})$ 真阳性率是 ROC 曲线中的 y 轴。
欠拟合	由于模型未捕捉到训练数据的复杂性，所产生的模型的预测能力较差。许多问题可能会导致欠拟合，包括： <ul style="list-style-type: none">训练错误的特征集。训练时间太短或学习率太低。训练所采用的正规化率过高。在深度神经网络中提供的隐藏层太少。
不带标签的示例	包含特征但不带标签的示例。不带标签的示例为要推断的输入。在半监督式和无监督式学习中，在训练过程中使用了不带标签的示例。
无监督式机器学习	训练模型以查找数据集中的模式，通常是不带标签的数据集。 无监督式机器学习的最常见用途为将数据聚合为类似示例组。例如，无监督式机器学习算法可根据音乐的各种属性将歌曲聚合成一起。生成的群集可成为其他机器学习算法（例如，音乐推荐服务）的输入。聚类在难以获得真实标签的领域可能会有所帮助。例如，在反滥用和反欺诈等领域，群集可以帮助人们更好地理解数据。 无监督式机器学习的另一个示例为主成分分析 (PCA)。例如，通过在包含数百万个购物车内容的数据集上应用 PCA，可能会发现，包含柠檬的购物车经常也包含抗酸剂。 与监督式机器学习进行比较。
验证	作为训练的一部分，用于使用验证集评估机器学习模型质量的过程。由于验证集与训练集不相交，因此验证有助于确保模型的性能在训练集之外得以推广。 与测试集形成对比。
验证集	用于验证的数据集子集，该子集与训练集不相交。 与训练集和测试集形成对比。
Weka	一组用于数据挖掘任务的机器学习算法。它包含用于数据准备、分类、回归、聚类、关联规则挖掘和可视化的工具。

版本历史

表 13. 文档版本历史

日期	版本	变更
2019 年 1 月 28 日	1	初始版本 增加了 MIPI I3C SM 接口 更新了第 1 节 LSM6DSOX 中的机器学习内核 更新了 1.1 节 输入 更新了 1.2.1 节 滤波系数 更新了特征 增加了图 6。峰到峰 更新了 1.3.5 节 过零检测至 1.3.12 节 最大值 增加了图 7。过零信号 增加了图 8。正过零信号 增加了图 9。负过零信号 更新了图 10。峰值探测器 更新了图 11。正峰值探测器 更新了图 12。负峰值探测器 增加了 1.3.13 节 特征选择 更新了 LSM6DSOX 中的 1.4 节 决策树和 1.4.1 节 决策树限制 更新了图 20。Weka Classify
2019 年 8 月 13 日	2	更新了图 21。决策树格式 更新了 LSM6DSOX 中的 1.5.1 节 元分类器限制 更新了 1.6 节 有限状态机接口 更新了 2.1 节 Unico GUI 更新了 2.2 节 决策树生成 更新了图 26。特征配置和图 27。ARFF 生成 更新了 3.1 节 振动监测 更新了 3.2 节 运动强度 更新了 3.3 节 6D 位置识别 更新了 3.4 节 智能手机应用程序的活动识别 更新了图 40。健身活动识别决策树 增加了第 3.6 节 示例汇总 更新了附录 A Weka 增加了附录 B RapidMiner 增加了附录 C Matlab 增加了附录 D Python 增加了附录 E 词汇表
2019 年 10 月 3 日	3	增加了 图 1. 监督式机器学习方法 更新了 图 23. 配置过程

目录

1	中的机器学习内核 LSM6DSOX	2
1.1	输入	3
1.2	滤波器	5
1.2.1	滤波系数	6
1.3	特征	9
1.3.1	均值	10
1.3.2	方差	10
1.3.3	能量	10
1.3.4	峰到峰	10
1.3.5	过零信号	10
1.3.6	正过零信号	10
1.3.7	负过零信号	11
1.3.8	峰值探测器	12
1.3.9	正峰值探测器	12
1.3.10	负峰值探测器	13
1.3.11	最小值	13
1.3.12	最大值	14
1.3.13	特征选择	14
1.4	决策树	15
1.4.1	中的决策树限制 LSM6DSOX	16
1.5	元分类器	17
1.5.1	中的元分类器限制 LSM6DSOX	17
1.6	有限状态机接口	17
2	机器学习内核工具	18
2.1	Unico GUI	18
2.2	决策树生成	21
2.3	配置过程	23
3	决策树示例	30
3.1	振动监测	30
3.2	运动强度	31

3.3	6D 位置识别	31
3.4	智能手机应用程序的活动识别	33
3.5	健身活动识别	36
3.6	示例汇总	37
附录 A	Weka	38
附录 B	RapidMiner	45
附录 C	Matlab	49
附录 D	Python	50
附录 E	词汇表	51
版本历史	55	
目录	56	
表一览	58	
图一览	59	



表一览

表 1.	机器学习内核输出数据速率	2
表 2.	滤波系数	6
表 3.	滤波器系数示例	7
表 4.	特征	9
表 5.	决策树结果	15
表 6.	决策树中断	16
表 7.	中的决策树限制 LSM6DSOX	16
表 8.	元分类器示例	17
表 9.	中的元分类器限制 LSM6DSOX	17
表 10.	智能手机配置的活动识别.	33
表 11.	健身活动识别配置	36
表 12.	示例汇总	37
表 13.	文档版本历史	55



图一览

图 1.	监督式机器学习方法	1
图 2.	中的机器学习内核 LSM6DSOX	2
图 3.	机器学习内核块	3
图 4.	MLC 输入（加速度计）	4
图 5.	MLC 输入（陀螺仪）	4
图 6.	滤波器基本元素	6
图 7.	峰到峰	10
图 8.	过零信号	10
图 9.	正过零信号	11
图 10.	负过零信号	11
图 11.	峰值探测器	12
图 12.	正峰值探测器	12
图 13.	负峰值探测器	13
图 14.	最小值	13
图 15.	最大值	14
图 16.	决策树节点	15
图 17.	Unico GUI	18
图 18.	机器学习内核工具 - Data Patterns	19
图 19.	机器学习内核工具 - 配置	20
图 20.	Weka Preprocess	21
图 21.	Weka Classify	22
图 22.	决策树格式	23
图 23.	配置过程	24
图 24.	将结果分配给数据模式	25
图 25.	机器学习内核配置	25
图 26.	滤波器配置	26
图 27.	特征配置	26
图 28.	ARFF 生成	26
图 29.	ARFF 文件	27
图 30.	结果和决策树配置	27
图 31.	元分类器和设备配置	28
图 32.	Unico 载入配置	28
图 33.	Unico 数据窗口	29
图 34.	Unico - 机器学习内核源寄存器	29
图 35.	振动监测决策树	30
图 36.	运动强度决策树	31
图 37.	6D 位置	32
图 38.	6D 决策树	32
图 39.	智能手机决策树的活动识别	34
图 40.	Weka 交叉验证	35
图 41.	健身活动识别决策树	37
图 42.	ARFF 示例	38
图 43.	Weka GUI Chooser	39
图 44.	Weka Explorer	40
图 45.	Weka Explorer - 属性	41
图 46.	Weka 分类	42
图 47.	Weka 分类 J48	42
图 48.	Weka J48 分类器参数	43
图 49.	正确分类实例	44
图 50.	混淆矩阵	44
图 51.	RapidMiner Studio - 打开文件	45
图 52.	RapidMiner Studio - 读取 ARFF	46



图 53.	RapidMiner Studio - 设置角色	46
图 54.	RapidMiner Studio - 决策树运算符	47
图 55.	RapidMiner Studio - 决策树图	47
图 56.	RapidMiner Studio - 决策树说明	48

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和/或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 标志是 ST 的商标。关于 ST 商标的其他信息，请访问 www.st.com/trademarks。其他所有产品或服务名称是其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。

© 2019 STMicroelectronics - 保留所有权利