# 异常检测（空气处理单元） - 模型生成

空气处理单元（AHU）用于调节和循环空气，是供暖、通风和空调（HVAC）系统的一部分。它吸入外部空气，调节后通过供风机将其作为加热或冷却的新鲜空气输送到建筑物内。控制因素包括外部空气温度、内部空气温度和房间或楼层的占用情况。在智能校园环境中，通常每个建筑的每层楼都有一个AHU。为了确保HVAC系统的高效运行，维护这些单元至关重要。AHU的故障通常由空气流量中断引起，并被标记为供风机故障。

我们将使用[SAS Visual Data Mining and Machine Learning (VDMML)](https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4\_3.5&docsetId=casactml&docsetTarget=casactml\_svdatadescription\_details.htm&locale=en)中封装的支持向量数据描述（SVDD）算法，并使用SAS事件流处理（ESPPy）模块进行部署。CAS会话用于调用深度学习以训练新模型，实时检测流数据中的异常。

关于此用例的更多资源可以在[SAS GitHub页面](https://github.com/sassoftware/iot-anomaly-detection-hvac)找到。

### 0. 设置环境

首先，导入运行此笔记本所需的包。

```python

from swat import \*

```

### 1. 启动SAS Viya CAS会话

CAS会话用于调用SAS Visual Data Mining and Machine Learning (VDMML)以训练新模型。

```python

cashost = 'localhost'

casport = 5570

sess = CAS(cashost, casport, caslib="casuser")

sess

```

### 2. 数据准备

定义生成模型期间使用的数据，并导入表操作集。

```python

indata\_dir = "/demo/Event\_Stream\_Processing/data"

indata1 = "ahu\_train"

indata2 = "ahu\_scr"

sess.loadactionset(actionset="table")

```

### 3. 将数据加载到CAS

数据每十五分钟从两个AHU中捕获一次，跨度为六个月。数据分为两部分：1. 训练数据 - 表示正常运行状态，收集时间为1.5个月；2. 评分数据 - 包含在4个月内收集的异常行为。训练和评分数据集包括传感器值，如混合空气温度、返回空气温度、冷冻水阀门状态、管道压力、供风机速度等。

```python

if not sess.table.tableExists(table=indata1).exists:

tbl = sess.upload\_file(indata\_dir + "/" + indata1 + ".csv", casout={"name": indata1})

if not sess.table.tableExists(table=indata2).exists:

tbl = sess.upload\_file(indata\_dir + "/" + indata2 + ".csv", casout={"name": indata2})

```

### 获取表信息

```python

sess.tableinfo()

```

### 4. 导入SVDD操作集

```python

sess.loadactionset('svdd')

```

### 5. 训练SVDD模型

SVDD算法是一种一类分类技术，适用于只有大量一类数据，而关于其他类数据很少或没有的应用场景。欺诈检测、设备健康监控和过程控制是一些应用领域，大多数数据属于一类。

最简单的形式中，SVDD模型通过围绕一类训练数据构建最小半径超球体来获得。超球体提供了训练数据的紧凑球形描述。该训练数据描述可用于确定新观测是否类似于训练数据观测。计算任何新观测到超球体中心的距离，并与超球体半径进行比较。如果距离大于半径，则观测被指定为异常。使用SVDD公式中的核函数可以提供更灵活的训练数据描述。这种描述是非球形的，符合数据的几何形状。PROC SVDD仅实现了灵活的数据描述。

我们使用基于SVDD的K图来确定AHU的异常行为。K图是一种非参数多变量控制图，用于统计过程控制，也可用于监控设备健康和操作数据。它分为两个阶段实现：第一阶段，收集正常操作过程的观测数据，并使用这些数据训练SVDD模型，获得阈值r平方值。

#### 第一阶段：模型训练

运行svDataDescription.svddTrain操作集训练数据

```python

sess.svDataDescription.svddTrain(bw=94,

solver='actset',

inputs=[{"name":"SUPPL\_FAN\_SP"},{"name":"DIS\_AIR\_TEMP"},{"name":"DUCT\_PRESS\_ACTV"},{"name":"MIXED\_AIR\_TEMP"},

{"name":"RTRN\_AIR\_TEMP"},{"name":"MAX\_CO2\_VAL"},{"name":"CHW\_VALVE"},{"name":"CHW\_VALVE\_POSIT"}],

id=["AHU"],

savestate={"name": "svdd\_ahu", "replace": True},

output={"casout": {"name": "sv", "replace": True}},

table={"caslib": "casuser", "name": "ahu\_train"}

)

```

### 6. 使用ASTORE对SVDD模型进行评分

在第二阶段，此正常操作模型被用于异常检测。对于每个新观测，计算其距离值并与阈值r平方值进行比较。距离值大于阈值r平方的观测通常表示过程中的异常情况。

加载astore操作集

```python

sess.loadactionset('astore')

sess.score(

table='ahu\_scr',

out='ahu\_scored',

rstore='svdd\_ahu'

)

```

### 7. 生成分析存储文件

此ASTORE文件可用于ESPPy进行实时异常检测。

```python

store = sess.download(rstore='svdd\_ahu')

with open('/user/my\_code/ahu\_svdd.astore', 'wb') as file:

file.write(store['blob'])

sess.astore.describe(rstore='svdd\_ahu')

sess.tableinfo()

```

### 8. 清理项目

终止CAS会话。

```python

#sess.close()

```