基于孤立森林算法的智能水务分析模型

摘要

如何通过有限的水流量数据来判断一个区域的供水系统是否发生故障是一个重要的问题，但是为了解决这个问题需要面对着许多挑战，比如模型的通用性、模型的性能、模型受噪声的影响等，可以说这些异常数据的检测至关重要，如何从大量水流量数据中快速找到异常数据是解决这些问题的关键所在，本文将探索并建立合适的模型来进行异常数据的检测，讨论提升模型性能的可能性。

对于第一个任务：首先利用四分位距的方法，判断水流量数据的分布，水流量数据的分布不符合正态分布，因此不采用 Z-score 来判断异常数据。接着画出箱线图来具象化数据，通过图形可以找到数据的模式、水流量数据的特征，箱线图可以反映上下四分位距，根据四分位距可以确定正常数据的范围，确定判定异常数据的标准，初步找出超出范围之外的异常值，并进一步把他们分为温和异常值和极端异常值。

但是箱线图只是能帮助初步分析数据，不能作为通用模型，箱线图依旧存在着无法准确检测异常数据的问题，因为箱线图有时会将很多正常的数据判断为异常数据。

对于第二个任务：为了建立合适的通用模型，文章接下来基于数据的特点进一步选取合适的算法，经过对数据模式的分析，抓住数据的特点，考虑了孤立森林算法，文章讨论了孤立森林算法是否适用于水流量数据的异常检测，并最终决定采用孤立森林算法来检测异常数据，后续的讨论中可以看到，孤立森林算法检测的异常值是符合上述异常值检测标准的。孤立森林算法是一种无监督学习算法，可以很好地快速处理大量数据，鲁棒性好，还能够将噪声的影响降到最低，在没有异常数据也能够进行训练。基于数据的特点，选择合适的数据集——异常数据较少、异常值差别较大进行训练。然后在所有的八个虚拟地区进行检测，得到结果并分析数据特点与数据模式。

对于第三个任务：本文希望衡量该模型的优劣，训练所得模型的好坏取决于训练集的划分和参数的选取。可以通过箱线图对数据进行的分析来划分训练集，然后在测试集上进行测试，通过测试结果来计算精确度和 F1-Score 等指标，用于评判该模型优劣, 然后通过对模型灵敏度的分析，指出应该如何优化该模型，可以怎样提升模型的性能。

关键词：IQR、非正态分布、箱线图、孤立森林算法

1 问题重述

1.1 问题背景

供水系统在我们的日程生活中至关重要，但是供水系统有时会发生各种各样的故障，从而导致漏水的发生，这是一个大问题。在这样的背景下，电磁流量计应运而生，用于测量流量以及监测漏水，一种方法是获取某一区域输入和输出水流量的插值加以评价。

如今已经有许多基于流量数据的分析方法，但是还有一些挑战存在，比较重要的三个：首先是需要设计一个通用模型来了解流量计的数据模式，然后是如何更加快速地检测流量异常，最后一个挑战是如何应对噪声的影响。[1, C1]

1.2 具体问题

您的团队需要设计一个模型来应对上述挑战，需要对给定数据进行清理，开发异常检测模型，并优化模型，数据是八个不同虚拟区域的输入水流量和输出水流量之差。具体的任务有：

- 分析数据模式，建立从检测给定数据中识别异常数据的标准
- 在数据分析结果的基础上，建立通用模型对八个区域进行异常值检测
- 在给定数据集上测试模型并解释建模和异常值检测的结果

解决这些问题的关键是找到一个合适的异常数据检测算法。

2 数据模式分析与异常检测标准的确立

2.1 数据概览分析

题目所给的数据来自于八个不同的虚拟地区，每个地区的流量差值是一个与时间相关的变量，这些值中既有正数，也有负数。大部分流量之差的绝对值都在 10 以内，如果发现绝对值过大，则可初步认为该数据是异常数据。流量数据随着时间的变化而变化，不同地区流量数据随时间变化的规律也各不相同。

首先判断各个地区的数据是否满足正态分布，采用四分位距的方法判断是否满足或者接近于正态分布，如果符合正态分布，则可以使用 Z-score 判断数据是否异常，如果不符合，则需要考虑其他的方法。

计算可得八个虚拟地区输入和输出水流量之差的四分位距：

region_1	region_2	region_3	region_4	region_5	region_6	region_7	region_8
3.802019768	1.108408668	1.2881612	0.678413944	4.157332959	4.746437346	6.514955518	1.616935081

可见并不满足正态分布的要求，因此不能采用 Z-score 判断数据是否异常。对于不满足正态分布的数据，可以使用箱线图进行异常数据的检测。

2.2 利用箱线图进行数据分析并检测异常数据

下面是根据八个地区数据所作出的箱线图1，可以初步比较直观地对数据进行分析，也可以看出哪些是异常数据：

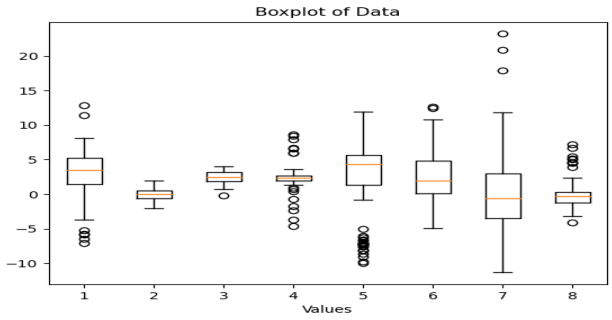


图 1: 八个虚拟地区的箱线图

从箱线图中，我们可以直观地看出每个地区的中位数、四分位数、异常数据以及数据的集中程度，箱线图较短的区域如 2、3、4、8 号地区数据集中程度就比较好。也可以用箱线图判断不同地区的数据趋向于哪种分布，比如区域三更趋向于正态分布，而其他区域的分布都有更多的偏移，既有左偏的分布又有右偏的分布，这些都可以直观地从箱线图中看出来。

2.3 异常检测标准的确立

已经使用了箱线图的办法，因此类似于正态分布中的 3σ 原则，可以借助第一、第三四分位数和四分位距 IQR 来对异常数据进行判断，对于每一个区域，找出他的第一和第三四分位数，然后先计算 IQR：

$$IQR = Q_3 - Q_1$$

然后计算内限范围：

$$R_1 = [Q_1 - 1.5IQR, Q_3 + 1.5IQR]$$

以及外限范围：

$$R_2 = [Q_1 - 3IQR, Q_3 + 3IQR]$$

处于内限范围以外的均为异常值，其中在外限以内的是温和异常值，在外限以外的是极端异常值。第 4、5、8 区域的异常值较多，第 4 区域的异常值最多，可以推测该地区的水流计发生了故障。

3 建立通用异常检测模型

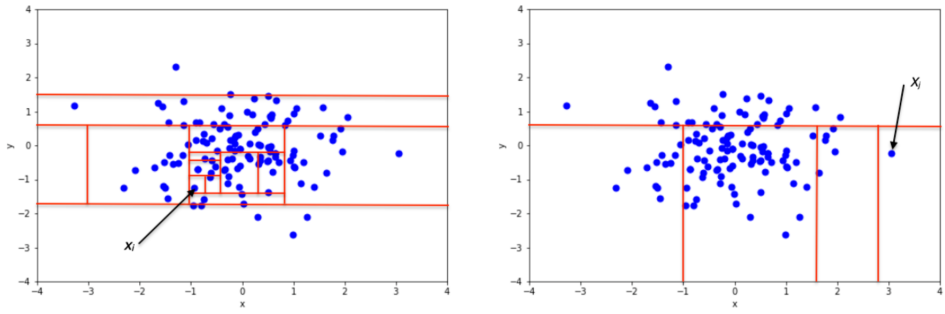
3.1 通用模型算法的选取与建立

通过箱线图进行分析以后，可以初步分析数据特征、找出异常数据，但是为了建立通用模型来检测异常数据，箱线图显然并不合适，箱线图只适合于反映数据的一些基本特征，在检测异常数据上不合适，比如区域 4 数据显然大部分都是正常的，然而箱线图却看出很多异常数据，这体现了箱线图的一些问题，那就是会检测出过多的异常数据。

通过上述的箱线图可知，每个区域异常值的占比都很低，且异常数据和正常数据的差别较大。因此可以采用孤立森林算法建立通用模型来进行异常值的检测。

孤立森林算法不需要对数据的分布做出假设，也不需要预处理数据，即使没有异常数据也能进行训练，这种算法大大降低了噪声的影响，训练所得模型可以快速处理大量数据并准确检测异常值 [2, iForest]

下面展示了二维高斯分布中应用该方法找出孤立数据 [3]



(a) (b)

图 2: 应用孤立森林算法的例子

图(a)反映了正常数据的划分情形，而图(b)反映了孤立点也就是异常数据的情形。

算法是通过构建多棵 iTree（isolation Tree）来实现的，通过随机选择数据中的某些特征，并在这些特征的范围内随机选择一个划分点，将数据分配到树的不同分支上，直到每个叶子节点中只包含一个或少量的数据点。由于异常点很少，所以它们往往需要更少的分割才能被隔离到树的较深层。因此，异常点在孤立森林中的路径长度通常比正常点短。由于每个数据点都可以用路径长度来表示，因此可以将路径长度作为异常度量，即路径长度越小，则数据点越可能是异常值。

路径长度的计算公式如下：

$$\mathrm{h}(\mathrm{x}) = \mathrm{e} + \mathrm{c}(\mathrm{T.size})$$

然后就可以计算异常分了，有时候计算异常分有不同方式，因此划分的方式也有所不同，但不会影响结果，比如 scikit-learn 中异常值由负数表示，正数意味着正常，下面是一种常规的归一化计算方法：

$$\mathrm{s}(\mathrm{x}, \mathrm{n}) = 2^{-\frac{\mathrm{E(h(x))}}{\mathrm{c(n)}}}$$

得分接近于 1 时，路径长度非常小，数据就会被孤立，因此较高的异常值路径长度较低。

3.2 训练集的划分与训练参数的设置

基于孤立森林法的特点，为了提高训练效果，提高模型的准确性、稳定性，应当选取异常数据较少地区的数据进行训练，另外，也要选取合适的参数。[4]

选取异常数据较少且异常数据差异较大的地区的数据作为训练集，选取区域 7 来进行训练，完成训练后可以在八个地区上进行测试。

在设定参数之前，先列举一下重要参数：

- n_estimators: iTree 的个数
- max_samples: 构建子树的样本数
- contamination: 异常数据所占的比例
- max_features: 每个子树的特征数
- bootstrap：采样是有放回还是无放回
- n_jobs: 作业数量
- random_state：训练的随机性
- verbose: 训练中打印日志的详细程度
- warm_start：是否重用上次调用的结果去 fit

参数设置考虑如下：
n_estimators，这个参数越大，算法的运行时间就越长，但是模型的性能也有可能提高。这里设置为 100，如果不满足性能要求可以继续提高。
max_samples，每个随机树使用的样本数量。这个参数可以控制随机树的分支程度。设置为”auto”，这样算法会使用全部数据样本。
max_features，每个随机树使用的最大特征数量。这个参数可以控制随机树的分支程度。设置为”auto”，这样算法会使用全部特征。
contamination，异常样本比例，提高该参数将使得标准更为严格，可能会使很多正常数据被判断为异常数据，因此设为 0.05 即可

- n_estimators = 100
- max_samples = ’auto’
- contamination = 0.05
- max_features = 1
- bootstrap = False
- n_jobs = None
- random_state = None
- verbose = 0
- warm_start = False

大部分均为默认参数设置，在这样的参数设定下，调用 sklearn 的 IsolationForest 模块进行训练。

4 模型的测试与评价

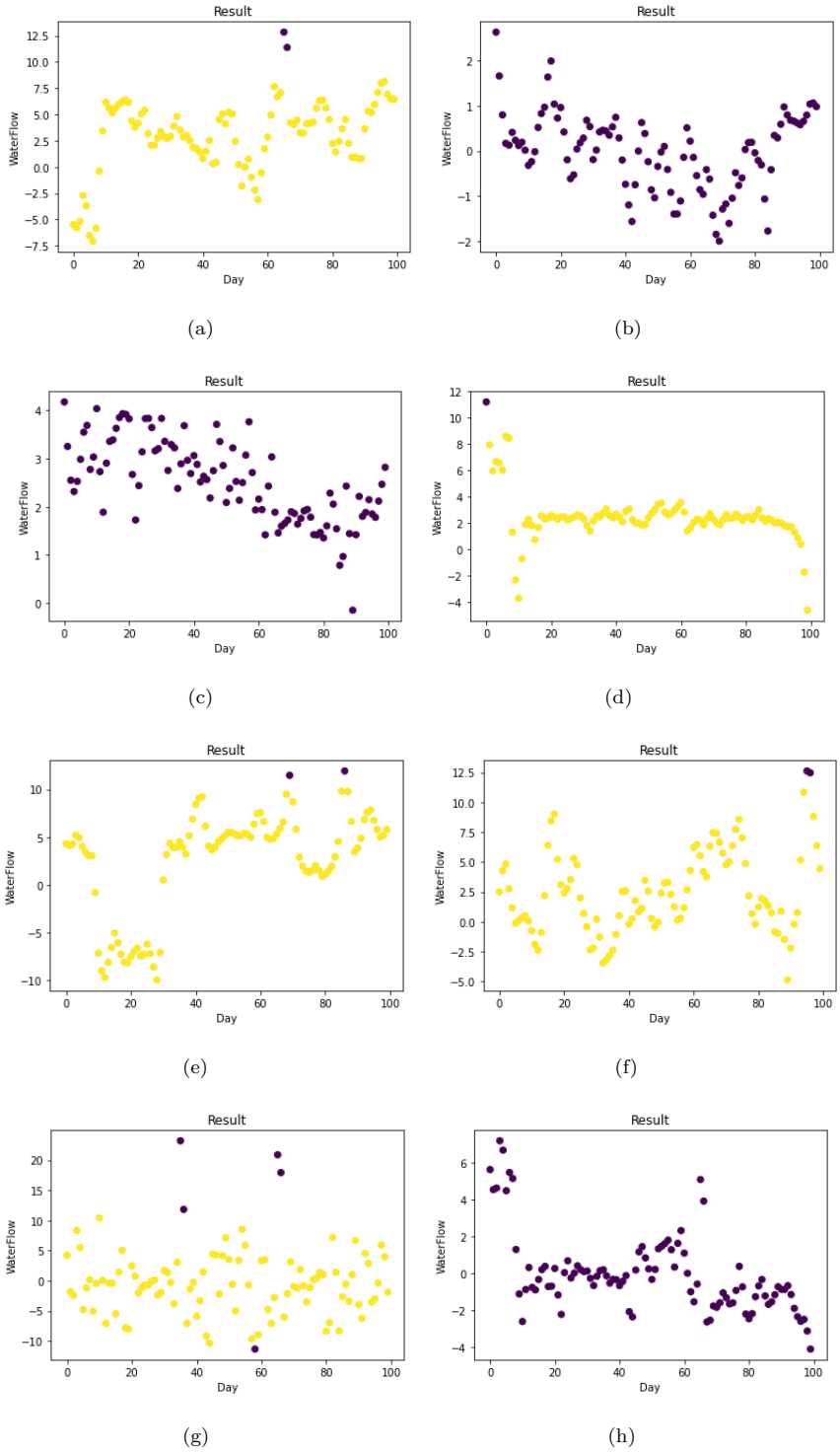
4.1 利用模型检测各个区域的异常数据

在区域 7 数据上完成数据的训练后，对每个区域进行检测，检测结果如下表所示：

region1	region2	region3	region4	region5	region6	region7	region8
12.8423647800	none	none	11.2105270500	11.4587055000	12.61496345	23.2090330700	none
11.3752904700				11.9137123000	12.4629461600	11.8275897600	
						-11.3403016100	
						20.8840092400	
						17.9267807800	

这个结果与箱线图的结果有较大不同，这是因为箱线图对数据的要求过于严格，以至于很多正常的数据都在箱线图上被判断成了异常数据。有些区域没有异常数据，但有些区域有较多异常数据，可能是供水系统出了问题，需要检修。

为了更直观地认识数据模式，下面给出了是每个区域检测结果的散点图，可以更加直观地看出数据的分布以及异常数据大致是如何被检测出来的。



(a) (b) (c) (d) (e) (f) (g) (h)

图 3: 孤立森林检测散点图

说明本文建立的模型所检测的异常数据还是少数，异常数据在所有数据中占比较低，这也是符合日常生活的，日常生活中异常事件不常发生，相比之下，前文的箱线图的结果就比较违背常理了——异常数据过多。

4.2 再次测试模型并评价各项指标

现在需要一些指标来对模型进行评价，根据应用场景要求，模型完成的实际上是一个分类的任务，可以用精确度来衡量，但这不够全面，因为另一个指标——召回率，在这个模型的评价中也很重要，所以应当考虑综合了精确度和召回率的指标——F1-Score 来对模型进行评价。

于是再次测试，得到每个区域检测的精确度和 F1-score 指标 可见，该模型对于水流量异

	region1	region2	region3	region4	region5	region6	region7	region8
Accuracy	0.980000	1.000000	1.000000	0.990000	0.980000	0.970000	0.950000	1.000000
F1-Score	0.989899	1.000000	1.000000	0.994974	0.989899	0.984772	0.974359	1.000000

常数据的检测效果很好，无论是精确度还是 F1-Score 都比较理想。

4.3 模型的优化

虽然测试结果比较好，但是该模型可能仍然有一定的局限性，首先因为只有 100 天的数据，其次训练数据集的数据模式也并不一定是最合适的，如果有更好的数据集，也就是异常数据更少、异常值偏差更大的数据集，那么训练出来的模型检测将会更加准确。

另一方面，可以通过调整训练的参数来提升模型的性能，参数对于模型训练效果的影响是复杂的，只有不断试验，结合原理，不断改变参数的值，才能找到更合适的参数。有时可能会耗时更多，计算量更大，但有望提高模型的性能。

此外，这仅仅只考虑了水流量一个因素，故障的检测还可以考虑更复杂的物理因素，比如水压、流速等等，虽然可能会使模型更加复杂，但却能从本质上提高模型的精确度、灵敏度，带来质的飞跃。

参考文献

[1] 刘冠乔. 智慧水务信息化建设规划与实践. 水利电力技术与应用, 3(9), 2021.

[2] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[3] WiKipedia. Isolation forest. https://en.wikipedia.org/wiki/Isolation_forest.

[4] https://www.zhihu.com/people/lao-q-84. 超详细！孤立森林异常检测算法原理和实战（附代码）. https://zhuanlan.zhihu.com/p/492469453.

附录

I 程序源代码

绘制箱线图：

```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('workspace\data.csv')

fig, ax = plt.subplots()
ax.boxplot(data)

ax.set_title('Boxplot of Data')
ax.set_xlabel('Values')

plt.show()
```

训练并测试模型，绘制散点图

```
import numpy as np
import pandas as pd
from sklearn.ensemble import IsolationForest
import matplotlib.pyplot as plt

data = pd.read_csv('data.csv')

Xtrain = data['region_7'].values.reshape(-1, 1)
Xtest = data['region_1'].values.reshape(-1, 1)

model = IsolationForest(n_estimators=100, contamination=0.05)

model.fit(Xtrain)

y_pred = model.predict(Xtest)


outliers = Xtest[np.where(y_pred == -1)]

print('异常值：', outliers)

plt.scatter(np.arange(len(Xtest)), Xtest, c=y_pred, cmap='viridis')
```

```
23
24          plt.xlabel('Day')
25          plt.ylabel('WaterFlow')
26          plt.title('Result')
27
28          plt.show()
```

评价模型

```
1           import numpy as np
2           import pandas as pd
3           from sklearn.ensemble import IsolationForest
4           import matplotlib.pyplot as plt
5           from sklearn.metrics import accuracy_score, f1_score
6
7           data = pd.read_csv('data.csv')
8
9
10          Xtrain = data['region_7'].values.reshape(-1, 1)
11          Xtest = data['region_8'].values.reshape(-1, 1)
12
13          model = IsolationForest(n_estimators=100, contamination=0.05)
14
15          model.fit(Xtrain)
16
17          y_pred = model.predict(Xtest)
18
19          outliers = Xtest[np.where(y_pred == -1)]
20
21          print('异常值：', outliers)
22          accuracy = accuracy_score(y_true=np.ones(len(Xtest)), y_pred=y_pred)
23          print('预测准确率：', accuracy)
24          f1score = f1_score(y_true=np.ones(len(Xtest)), y_pred=y_pred)
25          print('F1分数：', f1score)
```

II 支撑材料文件列表

以下是建模过程中用到的文件合集

- 数据文件.xls：原始的数据文件
- sourcecode.rar：建模用到的代码文件、运行代码所需的数据文件和记录结果用的文件
- pictures.rar：写论文用到的图片，生成的图片