Cross-dataset Time Series Anomaly Detection for Cloud Systems

2022年10月3日 11:08

云系统的跨数据集时间序列异常检测

摘要

软件应用程序作为在线服务部署到云计算平台，保证可用性，检测云系统的异常很重要。

云监测数据多样化，没有足够的标记数据建立准确的异常检测模型。因此提出跨数据集的异常检测。

提出跨越数据集的异常检测：在现有的标记数据集上训练异常检测模型来检测新的未标记的数据集中的异常。
（但是监督学习不就是这样吗，学习带标签的数据，进而判断未标记的数据的标签？）
方法称为ATAD（主动转移异常检测）：整合了转移学习和主动学习。

转移学习：应用于将从源数据集学到的知识转移到目标数据集；
主动学习：应用于确定来自无标签数据集的一小部分样本的信息标签。

ATAD在跨数据集的时间序列异常检测中有效。

一.简介

把服务部署在云平台。
为了保持服务的高可靠性和可用性，从大量的云监测数据中准确和及时地检测出异常情况是非常重要的，但是也很困难。

由于云系统的特点，实践中异常检测会遇到七大挑战。
一个大型的云系统由各种服务组成，每个服务都与一些检测数据有关。对某些类型的数据，异常的特征在许多服务当中是一致的，对于其他类型的数据异常特征可能因服务的不同而不同。
比如90%的CPU利用率对计算密集型服务来说是正常的，对其他服务来说是异常的，因此简单的基于阈值的异常检测器很难在各种服务中表现良好。

许多基于机器学习的异常监督方法被提出，包括监督的和无监督的方法。
实际的云环境中，标记的数据很少，对检测性能要求很高。
无监督学习确实可以处理大量的数据，但是性能相当低。
监督学习确实可以有更高的准确度，但是数据很多而且多样，如果用人工标注异常情况，非常耗时。

因此提出了ATAD，可以实现云系统的跨数据集异常检测：在现有的有标签的数据集学习，应用到一个没有标签的数据集上进行异常检测。
比如从公共数据集中学习检测器，然后将其应用于从真实世界系统中收集的未标记的数据集。

ATAD包括两部分：

1）转移学习：

从标记的时间序列数据中学习到共同的异常行为转移到大量未标记的目标数据集。利用不同数据集的共性，减少目标数据集的标记工作。

2）主动学习：

在目标数据集中标记少量选定的样本来提高检测性能。

在转移学习中，识别了云监控数据的多个特征，通过聚类（无监督学习）选择一个适当的现有标签数据子集作为子源域，应用CORAL算法缩小源域和目标域之间的特征差异。

在主动学习中，利用UCD（不确定性-背景多样性）方法来推荐需要标记的信息数据点。

被标记的点用来重新训练由转移学习训练的分类器。

目的就是最大限度地减少标记工作，尽可能提高检测器的性能。

ATAD显示出比现有方法更高的准确性

二.背景和动机

云服务供应商需要快速和准确的异常检测。

异常检测指的是识别出罕见的项目、事件或观察结果，这些东西和大多数数据有很大不同。

云中的异常检测通常在云监控数据（比如KPI、性能计数器、CPU利用率、虚拟机停机时间、系统工作量等）上进行。

云监控数据通常以时间序列的形式呈现。

大规模云服务系统中异常检测的挑战：

1）异常的多样化特征：大规模云服务系统，不同的场景和组件对异常的容忍程度不同。

那很自然地会想到，为每个使用场景和系统组件设置准确地异常阈值，但是这时非常困难的。

2）时间序列数据中的异常检测：云监控数据是大规模的时间序列数据（维度非常高），具有节奏性。很多常用的机器学习算法不能直接应用（因为这些数据不是独立、同分布），而且一些深度学习的方法可能还需要大量标记数据。

3）无监督学习的性能不理想：无监督的机器学习技术可以用于异常检测，通常是检查异常值与正常数据分布的偏差（**类似于聚类中离群的点?**）。但是效果并不理想。

4）缺少标签的监督学习：如果时间序列数据都有标记，那SVM、随机森林可以很好应用，但是标记数据集非常复杂，几乎不可能，因此限制了监督学习的应用。

三.建议的方法（ATAD，主动转移异常检测）

结合了转移学习、主动学习

大致流程：

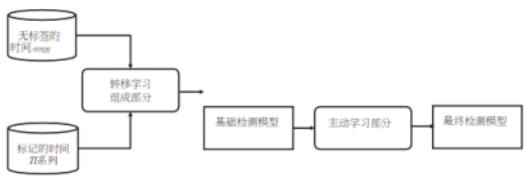

图1：ATAD的所有工作流程。

解释：有两组输入数据，一组是未标记的时间序列数据Tu，将对其进行异常检测；另一组是标记的时间序列数据Tl（从公共领域或云系统其他组件收集）（标记为异常或正常，就是一个二分类）。

转移学习：从原始数据集Tl中提取多个一般特征，形成特征数据集Fl，在Fl上学习一个基础检测模型。
未标记的时间序列Tu也提取同样的特征，形成Fu。
主动学习：通过不确定性和上下文多样性（UCD）策略，从Fu中选择少量信息样本进行标记。
用标记后的数据重新训练基础检测模型。经过T轮主动学习，最终得到异常检测器。

1.转移学习组件：

很多机器学习方法假定有标签的数据和无标签的数据分布相同。（就是在同一问题的数据集下分别训练和测试）

但是转移学习允许训练和测试中领域、任务、分布不相同。

云系统当中的异常检测问题直接进行迁移学习不可行。需要考虑以下因素：

1）云监测数据以时间序列的形式呈现，不是独立的数据，是有时间相关性的数据点。时间序列的异常模式具有背景相关性。（个人理解：不是简单的数值检测，而是和上下文、所处环境有关的）所以提取特征的时候需要保留上下文信息。

2）对一个时间序列来说，在什么样的粒度下进行迁移学习？

粗粒度：整个时间序列、子序列

细粒度：分散的时间点

粗粒度的异常检测不利于区分异常特征，同时定位和检索原因困难，因此选择细粒度。

3）迁移学习要求源域和目标域要有基本的相似性和类似的特征，需要过滤掉与目标域不相似的源域样本。

（1）特征识别

ATAD的特征工程将时间序列Tl中的每个数据点转换成一组特征Fl，这些特征可以捕捉到该点周围的背景和时间信息。

特征分为三组：统计特征、预测误差特征和时间特征。

计算特征之前，使用DFT估计最主要频率的周期p，不同的周期决定了以下过程中使用的滑动窗口的不同大小。

统计学特征：统计特征描述了时间序列中每个数据点周围的一些基本特征。有利于检测出违反基本特征的异常情况。比如在计算密集型服务中，如果某个时间窗口的平均CPU利用率趋于地下，这可能就是一个指标，表明其上的部分计算过程可能意外停止了。

统计学特征都是在一个周期p的时间窗口当中计算出来的。

预测误差特征：使用一组由时间序列预测产生的误差度量为特征。直觉是：如果当前点的值偏离预测结果，就更有可能出现异常。

使用集合模型（一些常用的模型）进行预测。

RMSE（均方根误差）赋予不同预测方法以不同的权重。

$$\hat{Y}_t = \sum_{m=1}^{M} \frac{\hat{Y}_{m,t}}{M-1} \cdot \left(1 - \frac{RMSE_{m,t}}{\sum_{n=1}^{M} RMSE_{n,t}}\right) \qquad (1)$$

Table 2: Metrics used as forecasting error features

Features	Formula	Description
ME	$\frac{\sum(Y_i-\hat{Y}_i)}{N}$	Mean Error.
RMSE	$\sqrt{\frac{\sum(Y_i-\hat{Y}_i)^2}{N}}$	Root Mean Squared Error.
MAE	$\frac{\sum\lvert Y_i-\hat{Y}_i\rvert}{N}$	Mean Absolute Error.
MPE	$\frac{1}{N}\cdot\sum\frac{Y_i-\hat{Y}_i}{Y}$	Mean Percentage Error.
MAPE	$\frac{1}{N}\cdot\sum\frac{\lvert Y_i-\hat{Y}_i\rvert}{Y}$	Mean Average Percentage Error.

时间性特征：一般来说，系统指标的急剧变化可能是反常的。例如磁盘的I/O流量急剧下降可能是由磁盘阵列的硬件故障引起的。

为了理解时间序列数据随时间的变化，通过比较两个连续的时间窗口确定时间性特征，还计算了两个窗口的差异。（后面的间隔可以设置）

（Q：但是理论上这种急剧的变化是可以在预测误差特征体现的？）

取决于这个预测是不是根据历史数据了。

（2）源域和目标域之间的转移

为了在两个域当中转移知识，有必要缩小它们的差异。

考虑到异常检测任务的有效性和效率的要求，提出了结合基于实例和基于特征的转移学习。

首先，已经保证了源域和目标域的数据相似。

源域可能由各种时间序列数据组成，需要收集与目标域数据相似的时间序列数据。

基于实例的方法过滤掉不相似的源域样本。

对于源域TI，确定特征集FI，对FI进行K-means聚类，建立K个聚类，每个聚类是FI的没有重叠的子集，可以看作是一个子源域。

为了选择相似的样本，未标记的时间序列数据形成特征数据集Fu。

计算每个未标记样本与每个聚类中心的欧式距离，样本被分配到最近的子源域。

（难道我们需要测试的未标记数据的特征数据集也是不同的吗？）

还需要进一步从特征空间的角度看，因为特征的分布可能仍然有不同。

对每个聚类进行CORelation ALighment（CORAL）。

它能以无监督的方式对准二阶统计学，即源和目标特征的共变性。

$$\min_A \left\|A^T C_l^i A - C_u^i\right\|_F^2$$

A是转换矩阵，Cil是标记的特征矩阵，Ciu是未标记的特征矩阵，缩小差异。

最后得到转换后的新的子源域特征数据Fi。

最后一步，在每个子源域上训练一个基础的监督模型。随机森林或者支持向量机。得到K个独立的基本模型。

该论文的基础模型采用随机森林。

分配过程是线性的复杂性。

子源域之间是完全独立的，后续处理可以并行进行，提高异常检测的效率。

2.主动学习组件

迁移学习技术不足以对云中的各种时间序列取得满意的结果。

主动学习的重点是最大限度地减少用户的标签工作，提高预测模型的准确性。

采用UCD方法（不确定性和上下文多样性），推荐一些样本进行标记。

（1）不确定性

大多数主动学习方法以不确定性为原则选择样本进行标记，因为人们认为如果一个模型对一些样本分类结果的确定性比较低，那么标记这些样本会对基础模型有帮助。

本方法使用基础模型（随机森林）来估计未标记的数据是正常或者异常的概率，用如下公式计算不确定性：

$$Uncertainty = -|Prob(Normal) - Prob(Anomaly)| \quad (3)$$

（先预测一遍）不确定性越大越需要标记。

（2）上下文的多样性

样本的多样性是需要考虑的因素。有时候两个样本很相似，可能属于同一个异常模式，就不需要都进行标记了。

传统的多样性方法一般基于clustering。

云系统中，系统指标的时间序列通常是连续的，没有断点，因此，相邻的样本往往相似。

具体来说，我们按不确定性对所有样本进行排序，顺序扫描。如果我们扫描的一个新样本与候选集的另一个样本在一个context中，即二者彼此相邻，我们就认为它们的信息可能是相似的，忽略这个新样本。反之，则添加到候选集。

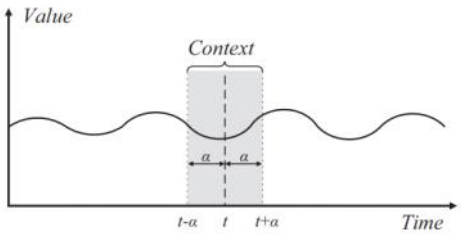

Figure 3: Context in time series

对于每个源域，在它自己的测试数据上进行主动学习，对不确定且不同的样本进行标记，标记的样本加入训练集，重新训练基础模型，在T轮之后，得到了最终检测模型。

3.ATAD的使用方法

转移学习和主动学习只是在训练过程中使用，ATAD训练完成，就得到一个分类器用于实际的异常检测任务。

将待检测的时间序列数据输入到特征提取器当中，像训练过程一样提取特征，之后这些特征被输入到训练好的分类器当中，得到异常概率。概率高于预先指定阈值（敏感参数）的点被预测为异常情况。

（所以源域是其他已经掌握的数据集，目标域是与待检测目标相似的数据集吗，如果是这样为什么会可以分到不同的聚类呢，可能是待检测的时间序列数据也是来自不同的服务吧。）

四.实验

本节通过一系列的实验来评估ATAD方法的有效性。

Q1：ATAD有效性如何?

Q2：迁移学习的效果如何?

Q3：主动学习的效果如何?

Q4：ATAD在检测一个公司基于公共数据集的本地数据集的异常方面有多大效果?

1.数据集和设置

使用两个公共的时间序列异常检测数据集NAB和Yahoo评估我们的方法。

NAB是一个新基准，用于评估流式实时应用中的异常检测算法。包含从不同领域收集的数据集AWS、Twitter等，每个数据集都有几个不同长度的时间序列。AWS包含不同的服务器指标，由Amazon Cloud-Watch提供.....

使用雅虎、AWS和Twitter数据集作为测试集（目标域）。

2.

使用F1 Score评估异常检测方法的准确性。

3.结果

（1）ATAD的效果如何?

从两方面看效果：一是检测结果，二是节省标签成本（相比于监督学习）。

对比的方法：iForest，K-Sigma，S-H-ESD，Random Forest。

一些参数设置.....

很明显，ATAD在所有数据集上都能取得比其他方法高的多的F1分数。

尽管监督学习（RF）比无监督学习的方法取得了更好的性能，但在给定相同数量的标签时，ATAD性能超过了RF。（因为有源域的帮助）

Table 5: Results of Comparative Methods

Dataset	Method	Precision	Recall	F1-Score
Non-Yahoo → Yahoo	iForest	0.3832	0.2183	0.2781
	K-Sigma	0.6499	0.3364	0.4433
	S-H-ESD	0.2779	0.6215	0.3840
	RF	0.8668	0.2075	0.3348
	ATAD	0.8847	0.4040	**0.5547**
Non-AWS → AWS	iForest	0.1523	0.0491	0.0743
	K-Sigma	0.6899	0.1992	0.3091
	S-H-ESD	0.5382	0.7100	0.6123
	RF	0.9999	0.6226	0.7674
	ATAD	0.9195	0.8142	**0.8637**
Non-Artificial → Artificial	iForest	0.3477	0.9006	0.5017
	K-Sigma	1.000	0.1730	0.2950
	S-H-ESD	0.7888	0.4568	0.5785
	RF	0.9182	0.9301	0.9241
	ATAD	0.9990	0.9850	**0.9924**
Non-Twitter → Twitter	iForest	0.4685	0.3087	0.3722
	K-Sigma	0.2608	1.0000	0.2608
	S-H-ESD	0.7481	0.4654	0.5739
	RF	0.7285	0.4811	0.5795
	ATAD	0.8769	0.6951	**0.7755**

为了证明ATAD在节省标记工作方面的优势，比较了ATAD和RF在相似的F1分数下标记样本的数量。监督学习模型需要比ATAD多3~10倍的标签才能达到相当的结果。
主要还是得益于知识的转移和UCD节省了标签数量。

（2）转移学习部分的效果如何?
从两个方面评估转移学习的有效性：所确定的特征是否有效，转移方法是否有效。

确定的特征的有效性:
本次转移学习的特征包括：预测误差、统计、时间特征，传统的迁移学习通常只用统计特征（平均数和方差）。
为了评估有效性，在四个数据集进行实验。比较单独使用统计特征和加入顺序感知特征。
实验表明，后者的性能好很多。因为转移学习需要缩小源域和目标域的差异，一定要充分利用时间序列的上下文特性。

随机森林可以评估特征的重要性。进行一个排序，可以发现预测误差特征和原始时间序列对异常检测的信息量更大，时间特征也有重要意义。但是每个数据集的重要特征不同，迁移学习的时候还是应该考虑所有特征。

转移方法的有效性:
转移方法：通过聚类创建多个独立子源域，进行CORAL算法转换子源域的特征（以接近目标域），希望以此减少目标域的标记工作。

对照的方法是不用转移学习，直接在目标领域应用主动学习，这种方法不需要辅助的标记数据作为转移学习的支撑，但是需要更多的标记工作建立基础模型。

实验表明，不转移学习需要更多的标签数，而且性能还略低于转移学习的，不过也比监督学习好，样本数更少。

（3）主动学习部分的效果如何?
UCD方法与传统的不确定性方法和随机选择方法，可以发现先UCD方法在所有数据集上取得的了最好的结果，证明了纳入时间序列背景多样性的实用性。
如果样本数越多，训练轮数越多，准确率越高，因为越贴近监督学习。

考虑的上下文的范围参数a也很重要，它是上下文多样性和不确定性的权衡，过大过小都不好。

（4）ATAD在检测一个公司基于公共数据集的本地数据集的异常方面有多大效果?
所有公共数据集作为源域，微软的大规模云系统的实际工业数据为目标域。
ATAD也比无监督和监督学习好。

五.对有效性的威胁（应该意思是，可能对有效性产生威胁的因素）

1.数据质量:
这项工作使用公共数据集进行评估，异常情况的标签域数据集一起提供，但是数据集可能也有少量的噪音，并且当前的数据量也比较有限。

2.标签的正确性:

ATAD中，主动学习要求用户手动标注一些数据，实验假设这些标签是正确的，但是可能人工标注的数据会出错。

3.数据泄露：

为了防止数据泄露，从测试集删除主动学习过程中标记的样本。忽略影响。

六.相关工作（已有的其他用于时间序列异常检测的方法）

大致可以分为监督方法、半监督方法、无监督方法和基于统计的方法。

1.无监督的方法：

不需要人工标记，假设正常实例的频率远远高于异常实例，并且异常实例偏离了正常的数据分布。

Ahmad等人提出一种无监督的在线序列记忆算法，称为分层时间记忆，用于检测流数据中的异常情况。

Xu等人提出Donut，基于变异自动编码器（VAE）的无监督异常检测算法。

2.监督的方法：建立一个正常和异常类的分类模型。

3.半监督的方法：训练数据只有正常类的标记实例。

4.基于统计的方法：K-sigma：样本值偏离相应的平均值超过样本方差的K倍，则视为异常点；还有极值理论的方法。

监督学习和半监督学习通常比无监督学习和基于统计和视觉化的方法好。
但是，标记数据集的成本很高，当数据集规模非常大的时候，很难应用于现实世界。
所以转移学习和主动学习技术用来解决这一重要问题。

思路总结：在云系统上，通常有大规模的数据，需要进行异常的检测以确保服务的可靠性。大量的数据用监督学习的话，需要贴标签，工作量极大，不现实；无监督的方法可以做，但是性能很差。于是提出了跨数据集的方法（迁移学习+主动学习）。
迁移学习首先要有一个已有标签的数据集，从时间序列、统计特征、预测误差三个维度归纳出特征，目标数据集也进行特征的归纳。然后对源域进行K-means聚类，再把目标数据集分进去。子源域利用线性代数的方法修改特征矩阵，尽可能贴近目标域。然后利用随机森林模型训练出基础模型。
主动学习首先利用UCD方法（不确定性，上下文无关性）尽可能高效地选出手动标注的目标数据，标注以后，重新训练基础模型，得到分类器。
最终的目标数据集也需要归纳特征，放进去分类即可。
效果很好。