通用可扩展的时间序列异常自动检测框架

**摘要**

该论文针对大规模时间序列数据提出了一种既通用又可以扩展的异常自动检测的框架。早期的异常检测在维护用户数据的连贯性以及保护企业免受恶意攻击方面扮演了很关键的角色。目前最先进的异常检测方法存在可扩展性、用例限制、使用困难和大量误报等问题。我们在雅虎的EGADS系统，使用了一组带有异常检测过滤层的异常检测和预测模型，以便在时间序列上进行准确和可扩展的异常检测。我们将我们的方法和其他具有不同时间序列特征的实时和合成数据异常检测系统进行了比较。我们发现我们的框架允许在精度和召回率方面为各种用例提高50-60%。数据和框架我们都已开源。特别地，开源的数据为这类异常检测工作提供了一个标准参考，这也是其开创的一个先河。

**介绍**

尽管计算机硬件和软件的快速发展带来了强大的应用程序，数以百计的软件bug和硬件错误在大规模的计算机集群中持续发生影响着用户的体验进而影响收入。不间断系统有严格的正常运行时间要求，对这些系统的持续监控至关重要。从数据分析的角度看，这意味着不间断地监控很多的时间序列数据从而发现隐藏的失败或异常。因为问题规模很大，人工监控这些数据从实际来讲不可行，这也导致我们使用机器学习和数据挖掘的方法进行自动异常检测。

一个异常，或者一个离群值，是偏离其他数据特别多的一个数据点。通常来说，大部分程序中的这些数据是被一个或多个运行的进程产生的，反映了系统的功能。当系统下面运行的进程表现出异常的行为，它就会产生离群值。对这些离群值快速高效的识别出在很多方面都很有用，包括：入侵检测，信用卡诈骗，传感器事件，医疗诊断，执法等等。

现行的自动异常检测遭受着很大数量的误报，这阻碍了这些系统的实际应用。用例，或者类型指定的异常检测模型在特定的应用中或许有较低的误报率，但是当时间序列的特征发生变化，如果不加适当的调整，这些模型的效果很差。6.3节展示了实际应用中‘只使用一种模型匹配所有异常’模型的缺点。

我们在雅虎上的系统叫做EGADS——“通用可扩展自动异常检测”，它能够准确、可扩展地进行异常检测。EGADS将预测，异常检测，报警功能分隔到三个相互隔离的组件，以保证其他人能够添加自己的模型到这些组件中。这篇文章主要讲述后两部分。

EGADS使用一组调节好的预设的模型减少误报，但仍能满足一般用户的需求。然而，更高级的用例需要系统捕获一些类型的异常但是过滤掉其他类型的异常。我们关注的异常或许在量级，严重程度或其他一些之前不知道的参数上有所不同，而且这些异常依赖于用例。因为如此，EGADS的报警模块使用机器学习的方法挑选出和消费者最相关的异常。

就我们所知，EGADS是第一个在异常检测方面融合了灵活性、准确性、可扩展性的综合系统。EGADS和异常检测基准数据集已经开源。基准数据集和系统的开源填补了异常检测基准数据集和框架的空白，能够帮助学术界和工业界合作研发更新奇的异常检测模型。在雅虎，EGADS被许多组用来分析每日数百万的时间序列数据。

第二部分我们介绍的是EGADS的结构，算法和报警模块会分别在第三、四部分介绍。之前的工作会在第五部分介绍。第六部分讨论了实验，第七、第八部分分别讨论了实际用例和结论。

**结构**

EGADS框架由三个主要模块构成：时间序列建模模块（TMM），异常检测模块（ADM），异常报警模块（AM）。

给定一个时间序列，TMM模块首先对时间序列建模，产生下一时刻的预测值，该值会被ADM和AM模块利用分别计算错误率以及筛选不要紧的告警信息。这些模块会在第三第四部分详细介绍。

EGADS被构建为一个框架是为了能够轻松地整合到现有的监控设施中。在雅虎，我们内部的雅虎监控服务（YMS）每秒钟产生数百万个数据点。因此，一个高精度可扩展的自动异常检测对雅虎来说至关重要。我们接下来会详细介绍整合到YMS的详细情况。

2.1系统集成

EGADS可以作为一个独立平台来运行，因此可以被大型系统当作一个库来使用。因此，设计一个EGADS和雅虎内部监控系统的接口十分重要。YMS一个关键的约束是规模，平台需要每秒钟需要评估数百万的数据点。因此，许多关于整合结构的决定都关注对实时数据分析的优化上。平台和YMS的整合如图一所示

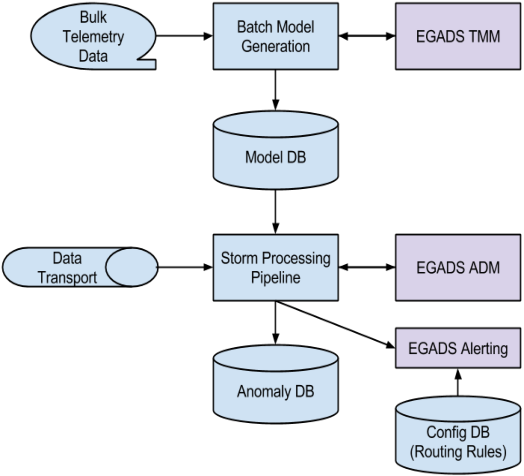


图 1 EGADS-YMS架构图

根据检测到的异常情况，需要多个支持组件来驱动操作。首先，所有的异常检测模型都是批量生成且实时应用的。批处理过程由以下三步构成：

1. 监测数据被大量存储在一个Hadoop集群中；
2. 一个批处理模型在这些数据上运行并对特定的时间序列建模；
3. 产生的模型被存储在模型数据库中

实时的数据流之后利用这些存储的模型：

1. 数据流入一个Storm[25]序列分析拓扑结构中。
2. 拓扑结构中的其中一个节点根据模型数据库中的模型调用EGADS的ADM模块对刚流入的数据进行分析。
3. 如果发现是一个异常，他就会被送到一个由组合规则和其他用例的特殊逻辑联合构成的次级规则流。
4. 根据这些规则，如果这一个异常是一个告警的事件，那么就会产生一个告警事件，并随之存储在状态数据库中，数据之后流向一个警报路由系统
5. 警报路由系统应用路由设置规则将警告发送到适合的系统管理员处。

2.2可扩展性

EGADS的监控用例需要在超过一亿个时间序列中每秒评估数百万个数据点。这在CPU负载、I/O和内存方面具有可扩展性。对一个数据点的评估需要尽可能地高效。这也意味着需要预先计算出尽可能多的模型。由于流量的增加，每次数据点到达时从磁盘读取模型是不现实的。这也表明模型需要被存放在内存中。为了控制占用的内存大小，模型需要尽可能地小。

一种优化方式是在多个相似的时间序列中共享模型。这在大型网站服务环境的情况下是适用的，因为应用总是被分解为类似服务器的水平层。这种优化方式能够减少模型数据库的内存的占用，批处理的工作负载和I/O。

另外一种可能的优化方式是研究自调整的模型；这种模型能够通过入站数据流基于在线学习而非周期性的批量生成。这种类型的模型或需要批量的进行初始化，但是总体上来看能够减少批处理负担。然而，从这些模型的实现上来看，它们可能要增加模型数据库的写操作，因为模型在不停地被改善。

另外一种优化方式包含了模型大小，训练速度和准确率之间的平衡。根据时间序列的特点，一个轻量快速的预测模型的准确度能够达到和一个复杂模型相似的程度。我们在6.2.2节评估了这些优化方式。

**异常检测算法**

在这一章，我们给出EGADS支持的异常检测算法的整体综述。现如今，EGADS能够检测出三种类型的异常：

1. 离群值：给定一个输入的时间序列x，一个离群值就是一个带有时间戳的数对<t,xi>，其中观测值xi和这个时刻的预测值差别十分大。
2. 变化点：给定一个输入时间序列x，一个变化点就是序列在时刻t的行为和t之前的行为大不相同。
3. 异常的时间序列：给定一个时间序列集合X = {x(i)}，一个异常的时间序列x(j) 就是一个和集合X中绝大多数行为不相同时间序列。

在接下来的各部分，我们将会给出EGADS针对上面提到的几种异常使用的异常检测方法的一般性概述。

离群值检测

离群值检测在许多监控系统中是十分重要的功能。因此，这篇论文的主要关注点也是离群值检测，除非特殊说明，否则论文中出现的异常都是指离群值。

EGADS为离群值检测提供了两类算法，我们会在后面介绍。

插入式方法

EGADS对时间序列的离群值检测的第一类方法叫做插入式方法。这些方法明确地对时间序列的正常行为建模，因此，在这个模型上一个大的偏离会被认为是一个离群值。为了给时间序列的正常行为建模，我们可以根据应用以及时间序列的特点使用很多时间序列模型和预测模型（例如ARIMA，指数平滑，克尔曼滤波，状态空间模型等等）。这也就是我们为什么称这种通用的策略叫做插入式的方法。应当注意的是，EGADS使用的所有时间序列预测模型也是我们系统中另一个特点；然而，因为论文的侧重点在异常检测，我们并不太给出这些模型以及EGADS在预测方面特点的详细介绍。

我们提出的插入式方法包括两种主要的模块：时间序列建模模块（TMM）和异常检测模块（ADM）。给定一个时间序列X={xt∈R：t>=0}，TMM模块给出t时刻xt的预测值，用ut表示。我们也将这个值叫做xt的期望值。TMM模块可以是一个基于训练数据给出预测值的机器学习模型，也可以是一个把一些在时刻t，xt应该如何表现的专业知识编码的基于规则的系统。在论文中，我们不对TMM做任何假设；也即是说，在我们提出的产生预测值ut的模型中TMM只是一个黑盒。在这种意义上，我们提出的框架是通用的，并不依赖于任何对时间序列建模的特殊框架。

给当一个预测值ut和一个实际观测值xt，ADM模块会计算出一些代表偏离的概念，我们称之为偏离度量（DM）。偏离的最简单的度量方式是预测残差，PEt = xt - ut.如果残差值落在设置好的阈值之外，一个警告会被触发。这种简单的方法在一些情况中适用，但对大多数情况来说这并不是一个好的策略，因为它不能捕捉到的相对误差。相对误差的定义如下：

REt = (xt-ut)/ut = xt / ut - 1

通过对相对误差进行阈值化，可以检测到异常，同时也可以规范化对期望值大小的依赖性。这些阈值的具体数值，实际上决定于异常检测模型的灵敏程度。不同的阈值设定方式会在第四章中介绍。尽管相对误差很通用且高效，但没有理由相信在给定时间序列的基础上进行异常检测的最优的方式。实际上，对一个给定的时间序列，最优度量的选择和TMM的表现一样，都取决于时间序列的特征。例如，如果我们正在处理一个十分规则的时间序列，而且我们有十分精准的预测模型，那么使用预测残差来进行异常检测已经足够，因为时间序列在这种情况下应该符合正太分布。在其他情况下，最优的度量或许是介于预测误差和相对误差的一种方式。因此，EGADS默认设置了一系列的偏离度量方式，系统使用者可以创造她自己的误差度量。这些误差度量方式和其他一些特征，如时间序列的特征，会被告警模块（AM）使用，在第四章中会介绍，用来学习消费者的行为方式以及过滤不重要的异常。

基于分解的方法

EGADS的第二类异常检测的方法基于时间序列的分解。特殊地，在时间序列分解的文献中，将时间序列分解为倾向、季节性、噪声三部分是一件很常见的事。通过监控噪声部分，一个人可以捕获离群值。更精确地说，如果xt点的噪声部分的绝对值超过了一个确定的阈值，便可以将xt当作离群值。

一个时间序列的分解可以在时域中通过平滑实现，也可以在频域中通过频谱分解实现。STL（基于Loess的季节性趋势分解）是一个使用Loess平滑实现分解的著名的方法。频域的分解也可以划分为带参数和不带参数的方法。对于带参数的方法来说，基于频谱分解的方法具有已知的参数化形式（例如傅里叶变换或者小波变换）然而对于不带参数的方法它们的基础是基于数据的。

**变化点检测**

变化点是那些在时间序列上的表现开始偏离它的预测值的点。变化点相对于多变的离群值的很大一个不同是变化点对应于更持久、长期的变化。文献中检测变化点的普遍的策略是在时间序列上移动两个并排的窗口，并计算两个窗口中时间序列之间的差异，以此作为偏差度量。在窗口中每个时间序列的行为通常使用数据的分布、图示、频率等表现在时间序列中的特征来进行建模。我们称这些方法叫做不受约束的方法，因为它们并不对时间序列的预测值做显式的假定。

在EGADS，我们现在已经采用了一种不同的方式，我们称作相对的或者说基于模型的方法。在这些方法中，时间序列的期望表现被3.1.1节提到的那些模型进行显式地建模。特殊地，我们结合3.1.1节提及的插件式的方法对输入的时间序列计算残差序列（或者说相对于模型预测值的偏差）。之后我们对时间序列的残差序列使用绝对变化点的检测方法来检测残差分布上的变化。我们已经使用了核密度估计来估计无参数的残差分布以及Kullback Leibler散度来测量分布上的变化。

我们认为基于模型的变化点检测方法比绝对的方法在实际应用中更加有用。这是因为变化点是有意义的，正如我们的模型无法解释某个时间点之后时间序列的行为一样。然而，从实际的角度来看，如果模型能够完全解释时间序列的表现即使在一个绝对变化点之后，我们没有任何理由把它当作一个变化点。换言之，变化点是相对于解释时间序列的行为下的模型而言的，这也就产生了相对变化点检测的方式。

**异常时间序列检测**

EGADS支持的另一类异常检测技术是检测异常的时间序列。一个平均偏离和其他时间序列相差很多的被定义为异常的时间序列T。假定所有的时间序列都是同种类的且来自同一个来源（即属于同一个集群的一部分），我们可以简单计算出时间序列（i）相对于其他时间序列的平均偏差。在EGADS中，我们现行的做法包括基于趋势和季节性，谱熵，自相关性，平均欧几里得距离等不同的时间序列特征将时间序列聚类到一系列的聚类中。在聚类之后，我们通过测量聚类中心和时间序列（i）之间或内部的偏差来进行聚类内部或聚类间时间序列的异常检测。对EGADS的这类异常检测，一个普遍的用例包括三分法。举例来说，如果一个网络工程师想要在数百万的时间序列中发现一个异常的服务器，使用前面提到的方法或许是不实际的，因为建模的过程是在每个时间序列的基础上进行的，而没有考虑其他行为的度量。这种异常检测方式的另外一种应用是查找相似的异常，这正好是上一种用例的相反。

**报警**

异常检测的最终目标是产生精准和及时的警告。EGADS通过两个步骤来完成这个目标，首先通过阈值选择产生一系列的异常候选，之后针对给定的用例筛选出相关的异常。

**阈值选择**

阈值选择的任务是针对异常检测模块（ADM）产生的偏离度量选择合适的阈值。EGADS现在基于Kσ偏离和密度分布应用了两种阈值选择的方法。

第一种方法是有参数的，它假定数据是拥有良好定义的平均值和标准差符合正太分布的。依赖于高斯分布，我们可以使用叫做“3-σ规则”的统计规则，这个规则假定99.73%的数据分布在均值的三个标准差之内。因此，根据Kσ中的数值K，我们可以确信在时间t观测到样本的概率。取决于所需的灵敏度水平，我们可以分别根据K=2或1的时候来观测一个给定的样本是不是在95.45%或者68.27%内。注意，这里的假定是我们的偏差度量是符合正态分布的。

第二种方法不是基于参数的，它在偏离度量不符合正态分布时很有用。基本的想法是在偏离度量分布中发现低密度的区域。一种方法是使用基于局部密度概念的局部异常因子（LOF）算法，其中局部由最近邻给出，其距离用于估计密度。通过将一个对象和其邻居的局部密度进行比较，我们可以识别相似密度的区域，和比邻居有相对低密度的点。这些将会被当作离群值。

**筛选**

筛选对异常进行最后一阶段的后处理，之后将异常传递给用户。尽管这些异常的候选集在作为筛选阶段的输入时在统计上十分重要，但并不是所有的都和特殊的用例相关联。例如一些人关注时间序列中的波峰，一些人关注波谷，还有一些人关注变化的点。EGADS提供一个简单直接的接口，使之能够让用户标注出时间序列中的异常区域。EGADS之后会使用这种反馈以及时间序列和模型特征来训练分类器使之能够预测是否一个异常ai和用户uj相关。EGADS使用的时间序列特征在表1中有介绍。

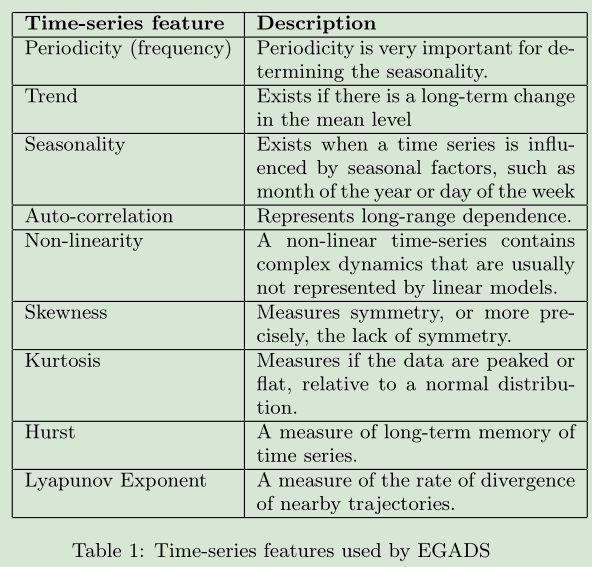


表 1

6.4节探究了针对特殊用例的筛选模块的性能。类似于EGADS中的其他模块，筛选模块从模块和特征的角度来看都是可扩展的。

图2描述了一个样本时间序列的特征概述。注意，以dc为起点的度量指标是从相邻的时间序列中获得的（例如在删去了趋势性和季节性之后）。在6.2节，我们查看了这些时间序列的特点是怎样影响模型的表现的。

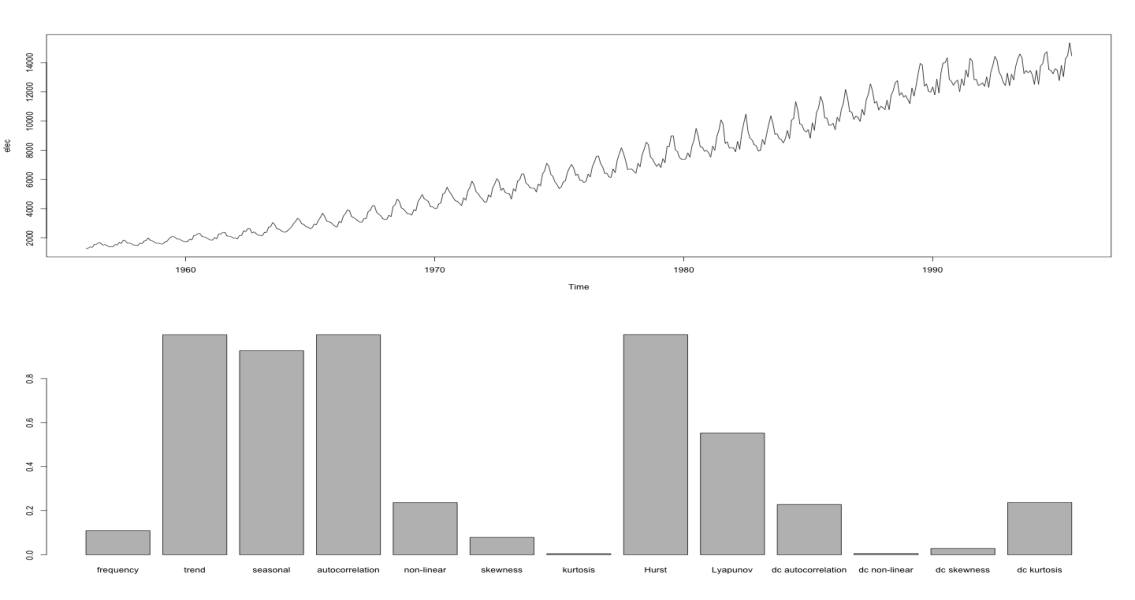


图 2一个时间序列的例子以及EGADS提取的特征。EGADS使用这些模型进行筛选和模型的选择

**相关工作**

在文献中有许多的异常检测方法。这些技术中有异常点检测算法也有改变点检测算法。在[24]中，作者提出一种基于假设检验的针对离群值检测的算法，这种算法在检测特定的离群值时特别准确。实际上，Twitter[26]使用[24]中的方法结合了时间序列的长期趋势来去除许多误报。Twitter的方法十分快并且有十分高的准确率和召回率，但是这种方法只是针对于Twitter的用例。在[27,15]中也有好多可用的开源的异常检测算法。

文献[13]中的作者也提出一种用于检测“变化点”或“层次变动”的异常检测方法。变化点不同于离群值点在于变化点反映的是时间序列的分布的改变。改变点通常发生在新产品功能或新平台发布的时间序列中。有许多开源的变化点检测算法，包括[14]在内。

在我们的实验中，一个特殊的异常检测算法通常只对某个特定的用例适用。就像文献[1]中作者所说的那样，异常通常会有一个比较高的异常评估分数，但是仅仅有一个比较高的异常评估分数并不足以说明这就是一个异常。更确切地说，正是管理员才规定了噪声和异常之间的区别。同样地，文献[4]对每种异常的检测方法提出一种简明的观点，引用了这样一个事实：只有一组异常检测模型最适合于特定的相关异常类别。因此，基于“一刀切”是异常检测世界中神话的观察结果，EGADS采用了一种策略，即用经过良好训练的异常检测模型集合和后处理用例特定的异常过滤阶段。然而，EGADS并不是唯一一个异常检测框架。Venkataraman等人提出一种“黑盒异常检测”框架，可以被用在多种数据源上。尽管提出的框架是通用的，它并不是全自动的，仍需要有相应级别的用户针对特定的应用设定特定的模型和度量。另外，这个框架也假定输入的数据是没有异常的，可这在许多真实世界的用例中并不现实。另一方面，Lan等人也针对大型系统提出异常检测框架，它是自动的，但通用性不够好，不能够很好的应用在一般的时间序列异常检测问题上。可EGADS提供了一种既灵活又高效的机制，能够实现通用、自动化和可扩展。而且，从工程的角度看，EGADS已经应用在雅虎的大型监控系统中。

**实验研究**