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。

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

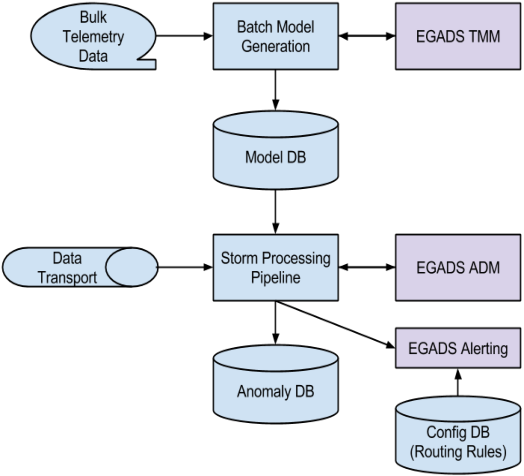


图 1 EGADS-YMS架构图

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。

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。