分布式数据流挖掘的研究进展

曲 武 隋海峰 杨炳儒 谢永红
(北京科技大学信息工程学院 北京 100083)

摘 要 随着通信技术和硬件设备的不断发展,尤其是小型无线传感设备的广泛应用,数据采集和生成技术变得越来越便捷和趋于自动化,研究人员正面临着如何管理和分析大规模动态数据集的问题。能够产生数据流的领域应用已经非常普遍,例如传感器网络、金融证券管理、网络监控、Web日志以及通信数据在线分析等新型应用。这些应用的特征是环境配备有多个分布式计算节点;这些节点往往临近于数据源;分析和监控这种环境下的数据,往往需要对挖掘任务、数据分布、数据流入速率和挖掘方法有一定的了解。综述了分布式数据流挖掘的当前进展概况,并展望了未来可能的、潜在的专题研究方向。

关键词 分布式数据流挖掘,数据流挖掘,数据流
中图法分类号 TP181 **文献标识码** A

Advances in Study of Distributed Mining of Data Streams

QU Wu SUI Hai-feng YANG Bing-ru XIE Yong-hong
(School of Information Engineering, University of Science and Technology Beijing, Beijing 100083, China)

Abstract With advances in communications technology and hardware equipment technologies, particularly the wide use of small wireless sensor devices, data collection and generation technologies have become more convenient and automated, organizations and researchers are faced with the ever growing problem of how to manage and analyze large dynamic datasets. Environments that produce streaming sources of data are becoming common place, such as sensor network, financial data management, network monitoring, Web log analysis and the communication data online analysis. In many application instances, these environments are also equipped with multiple distributed computing nodes that are often located near the data sources. Analyzing and monitoring data in such environments requires data mining technology that is cognizant of the mining task, the distributed nature of the data, and the data influx rate. We reviewed the current situation of the field and identified potential directions of future research.

Keywords Distributed mining of data streams, Data streams mining, Data stream

1 引言

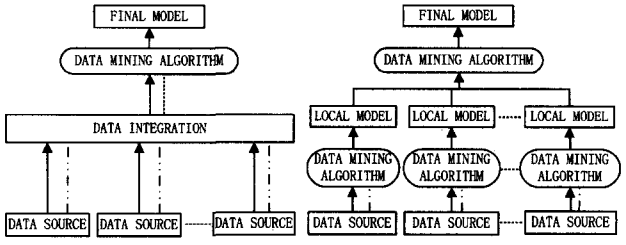
从20世纪60年代起,数据库技术迅速发展且得到了广泛应用。数据模型及其建模形式多样,从层次数据库、网状数据库、关系数据库、对象数据库,到对象关系数据库等;数据规模越来越大,数据处理和分析技术也越来越先进。目前应用最广泛的数据处理和分析技术仍然是比较成熟的数据库管理系统(Database Management System, DBMS)技术、数据仓库(Data Warehouse, DW)技术以及数据挖掘技术。然而上个世纪末开始在一些新型应用中出现的数据,却对传统数据处理和分析技术提出了新的挑战。这些数据包括传感器数据、网络数据、金融数据、事务日志、天文数据等,其种类还在不断地增加。这种数据的特点是数据持续到达,且速度快、规模宏大。为了从这些不断增加的数据中得益,类似数据挖掘这样的半自动化交互技术被用于处理和分析这些数据。由于处理过程在本质上是迭代的(人作为循环的一个环节),因此必须要求客户端查询的交互响应时间在合理的范围内,要满足这些标准的挑战往往是相当困难的(如下文所述)。在此背景下,数据流挖掘被广泛地应用于金融管理、网络监视、通信数据管理、Web应用、传感器网络数据处理等领域。

廉价的存储空间使我们能够存储大量数据,因此访问和管理这些数据就成为了一个性能瓶颈。而且一般情况下,单一的节点不能够存储如此大量的数据。因此,必须开发出适合和高效的数据访问、数据存储和通讯技术(如果数据源是分布的)。计算机网络、分布式计算技术的迅速发展,使得实时处理分布在网络不同节点上的数据流成为可能。此外,在连续流入数据的动态数据库情况下,数据挖掘变得更加复杂;数据的改变将会废止现存的模式或引入新的模式;从头执行算法还会导致大的计算量和I/O负载。这些因素导致了分布式数据流挖掘算法的产生和发展,并成为数据流挖掘研究的必

到稿日期:2011-02-15 返修日期:2011-04-19 本文受国家自然科学基金(60875029)资助。
曲 武(1981—),男,博士生,主要研究方向为知识发现与智能系统、分布式数据流挖掘技术与云计算,E-mail:quwu.ustb@gmail.com;隋海峰(1976—),男,博士生,主要研究方向为数据挖掘;杨炳儒(1943—),男,教授,主要研究方向为数据挖掘;谢永红(1970—),女,副教授,主要研究方向为数据挖掘。

然趋势。分布式数据流挖掘的研究在国际上刚刚起步，但它却展现了在军事、网络、金融等关系到国计民生的领域的广泛用途。

目前，许多系统使用集中模式进行多数据流挖掘[1]。在这种模式下，首先分布式的数据流被集中到中心节点，然后进行数据挖掘。集中式数据流挖掘系统模型如图 1(a)所示。这样的计算模型有以下几方面的局限性：第一，集中的数据流挖掘可能导致响应时间较长，而且分布式计算资源没有得到充分利用。第二，集中收集数据可能导致关键的通信链路负载过重。如果这些通信链路网络带宽有限，网络 I/O 很可能成为性能瓶颈。第三，在能量消耗约束环境，例如在传感器网络中，过量的数据通信会导致过多的能量消耗。为了缓解上述问题，研究人员提出了一个模型，这个模型充分考虑到了分布式数据源、计算资源和通信链路负载。这个分布式数据流挖掘系统模型如图 1(b)所示，可以与集中式数据流挖掘系统模型形成鲜明对比。在分布式数据流挖掘模型中，将数据集中到一个中央节点，分布式计算节点对最近邻的数据执行计算，局部模型必要时才与中心站点通信。这种架构具有以下优点：首先，通过分布式计算节点，更大程度地进行并行推导，从而缩短响应时间。其次，因为只有局部模式才需要通信，所以潜在地减少了通信量，提高了可扩展性，并降低了能源约束领域的功耗。



(a)集中式的数据流挖掘架构　　(b)分布式的数据流挖掘架构

图 1

本文主要介绍分布式数据流挖掘的相关算法、系统支持、系统应用，以及新兴的研究方向。首先介绍各种分布式数据流挖掘算法，如概要提取、Skyline 查询、分类、聚类、频繁项集挖掘和离群点检测；其次，简要介绍资源约束环境下的分布式数据流挖掘；再次，总结分布式数据流挖掘系统支持方面的研究工作；最后，对分布式数据流挖掘的发展趋势进行总结和展望。

2 分布式数据流挖掘算法

2.1 概要提取

在数据流处理系统中，由于数据量远大于可用内存，系统无法在内存中保存所有扫描过的数据，而数据流查询与挖掘经常会要求读取这些数据。为了避免代价昂贵的磁盘存取，系统必须在内存维持一个概要数据结构，以保留扫描过的信息。在分布式数据流环境下，研究人员对此问题也做了深入研究。

在文献[2]中，Bulut 等人提出了一种分布式环境下数据流渐增式的动态摘要算法，称为 SWAT(Stream Summarization using Wavelet-based Approximation Tree)。算法中，数据流的摘要以多种解析度被计算，然后它们一起诱导出一个基于小波的近似树。当从树根节点遍历到叶子节点时，贴近度会相应提高，近似树的空间复杂度，N 为当前数据流的大小。每个新数据值的处理成本摊销为 O(1)。这个算法在偏重查询模型下工作，即算法仅对当前到达的数据感兴趣。作者也考虑了在中心数据源站点以多种解析度概括一个数据流。客户端通过网络分布，并提出查询。中心站点计算的概要信息相应地缓存到客户端。数据流的访问模式(例如读和写)导致不同解析度的多个复制策略。当相应的数据读取速率提高时，每个复制策略就会增强；相应的数据写速率提高时，复制策略就会减弱。这种自适应的复制策略能够使总通信开销和站点之间通信量最小化。尽管这个获取概要的过程是集中式的，但通过聚集小波参数，用这种方法概要分布式站点的分布式数据流也是可行的。

在大量协同演化数据流中进行模式发现，这个问题在很多领域已经引起人们的注意。Papadimitriou 等人在文献[3]中提出一个 SPIRIT(Streaming Pattern Discovery in Multiple Time-series)方法。SPIRIT 方法是一个综合性的方法，可以通过发现流之间的关联性，更有效地提取数据流的概要信息。同时，这种方法占用更少的内存空间，而且内存需求和处理时间与数据流的长度无关，仅和数据流的数量成线性比例。这种方法是自适应的，而且完全自动。它可以动态检测输入数据流的变化(包括逐步变化和突发的情况)，同时可以自动测定隐含变量的数目。数据流的相关性和隐含变量的发现有多种用途，它们可以给用户提供简洁的概要信息，帮助进行快速预测和检测离群点，同时有利于插值和处理缺失值。虽然这个算法也是集中的，但它是针对分布式数据流，并在中心节点进行处理。不过，这个算法也可能被用来概要到达分布式站点的数据流。Chiky 等人在文献[4]中提出一个概要分布式数据流的通用框架。这个方法能够降低中心服务器的负载，并且通过最小化误差平方和的方式来优化每个输入数据流的采样率。这种优化是动态的，而且是根据数据流内容不断自适应的。

Babcock 等人在文献[5]中研究了分布式数据流上的最大 K 个值的查询算法(top-k 监控查询)。这个算法可以用来降低运行其它类型的监控查询时所带来的负载开销。作者认为，为了给查询提供支持而传输整个数据流是没有必要的。同时，他们提出另一种算法，该算法能够大幅度削减通信开销。在该算法中，远程流数据源维持的算术约束能够保证当前提供的 top-k 答案在用户指定的容差内仍然是有效的。而且，仅当算法约束不可用时，该算法才需要进行分布式通信。

2.2 Skyline 计算

Skyline 查询的研究始于 2001 年，最早由 Borzsonyi 等人提出[6]，主要关注在数据量很大、无法放入内存的情况下如何处理 Skyline 查询。具体地讲，Skyline 查询处理是指从给定的一个 D-维空间对象集合 S 中选择一个子集，该子集中的任意一个点都不能被 S 中的任意一个其他点所控制。所谓控制关系是指给定一个 D-维空间中的多个对象(集合 S)，如果对象 p 至少在某一维上优于另一个对象 q，而在其他维上都不比对象 q 差(p 优于或等于 q)，则说 p 能够控制 q。考虑如下股票推荐系统[7]：存在某股票数据库，股票数据具有风险和佣金两个属性。现某用户浏览该数据库，试图从最近 3 天内的股票中挑选一支风险和佣金均尽可能低的股票进行投资。Skyline 查询为这类应用提供了一个可行的解决途径，它返回

给定对象集中所有不被其他对象所支配的对象。这样，用户只需考虑属于 Skyline 集合的那些对象，不必关心被过滤掉的对象。Skyline 查询对于多约束决策支持、城市导航以及用户偏好查询等具有重要意义。

此前的大量工作[6,8,9]都专注于静态数据集上的 Skyline 计算问题。Borzsony 等人提出了 BNL(Block Nested Loops)和 D&C(Divide and Conquer)两种算法：BNL 算法采用迭代的方法将对象进行逐一对比，每次迭代产生一组 Skyline 对象；BNL 的优点是无需建立索引或把数据文件排序，可以应用到任意维空间；其缺点是依赖主存，当候选列表比主存还大时，需要将列表中的部分数据保存到临时文件中，导致 BNL 算法的多遍执行。D&C 算法则采用递归的方法，将整个数据集划分为若干个内存能容纳的子块，先分别计算各子块上局部的 Skyline，最后将局部结果合并，得到全局结果。D&C 方法在数据规模较小时具有较高的效率，对于大数据集，划分合并过程会产生大量的 I/O 代价。文献[10]将原始数据集编码成位图(Bitmap)，提出了一种基于比特位运算的改进方法，同时提出了一种基于索引(Index)的方法。这两种方法均能够在扫描全部数据集之前开始输出 Skyline 结果，且具有较高的运算效率。但是这两种方法需要预先对数据集进行编码和创建索引，预处理需要较大的计算和存储开销，且不适合频繁发生插入删除变化的数据集。在文献[11]中，Kossmann 等人基于 R*-tree 索引结构[12]提出的 NN 算法是渐进、公平的，而且能够接受外界交互信息，从而按照不同顺序生成 Skyline 集合，因此具有很好的灵活性。在文献[9]中，Papadiast 等人分析了 NN 方法在 I/O 和存储方面存在的不足并对之进行了改进，提出了依赖索引结构的 BBS(Branch and Bound Skyline)算法。在 BBS 算法中，对象采用 R-Tree 索引结构[13]来组织，然后采用分支定界的方法逐步输出数据集上的 Skyline，并在理论上证明了 BBS 算法是 I/O 最优的。在文献[8]中，Chomieki 等人分析了 BNL 算法，提出了 SFS(Sort Filter Skyline)算法。SFS 算法则采用先排序后比对的方法来计算 Skyline。

近几年数据流处理成为数据库领域的一个研究热点。文献[14,15]考虑了滑动窗口数据流上的连续 Skyline 计算问题。上述工作主要是在数据库领域考虑集中环境下的 Skyline 计算问题，其目标是利用空间索引或编码技术快速得到数据集的 Skyline，降低查询的 CPU、内存和 I/O 消耗。与集中式环境不同，分布环境下 Skyline 计算主要考虑如何减少网络通讯量。在文献[16]中，Balke 等人研究了分布环境下数据垂直分割时的 Skyline 查询处理技术。文献[17-19]等研究了 P2P 环境下的 Skyline 查询问题。文献[20,21]等研究了无线自组网(MANET，移动自组网；WSN，无线传感器网络)环境下的 Skyline 查询问题。但上述工作均基于传统数据库数据，没有考虑数据流的场景。在分布式数据流上 Skyline 查询方面的研究文献较少。文献[22,23]中，Sun 等人基于非共享策略，围绕着降低系统反应延迟与通信负荷的目标，提出了一种分两阶段渐进求解的分布式算法 BOCS(Based on the Change of Skyline)，并对算法的关键实现环节，如协调站点与远程站点间的通信、Skyline 增量的计算等进行了系统优化，使算法在通信负荷与反应延迟上达到了较好的综合性能，并从理论上分析证明，在所有基于非共享策略的算法中，BOCS 算法通信最优。大量的对比实验结果也表明，所提出的算法高效、稳定且具有良好的可扩展性。文献[24]中，Wang 等人为了降低分布式数据流上的连续 Skyline 计算过程中的通信开销，提出了基于远程过滤的思想并对相关理论基础进行了证明，描述了系统的体系结构并提出了两个过滤模型 v-Max 和 Distance。并且从理论分析和实验结果中证明了所提方法在某些数据分布情况下降低通信开销的有效性。

2.3 分类

Hulten 和 Domingos 在文献[25]中提出 VFDT 方法，利用 Hoeffding 不等式很好地解决了在数据流上进行单遍扫描获取高精度决策树的问题。VFDT(Very Fast Decision Tree)是一种基于 Hoeffding 不等式对数据流挖掘环境建立分类决策树的方法，它通过不断地将叶节点替换为分支节点而生成。其最主要的创新是利用 Hoeffding 不等式确定叶节点变为分支节点所需要的样本数目。Jin 和 Agrawal 在文献[26]中再次讨论这个问题，并提出了解决方案。他们的方法在获得与文献[25]同样精度的基础上，不但提高了分裂点的计算速度，而且减少了所需的样本量。但是，这两种方法都不适合处理分布式数据流。

在文献[27]中，Kargupta 和 Park 提出了一种分布式环境下的聚合决策树方法。每个决策树可以表示为一个数值函数，作者提出使用傅立叶变换来近似表示决策树，然后以此聚合这些树，并且在带宽限制的移动环境下传输。基于傅立叶的决策树表示方法以及数据流决策树构建算法，使该方法能够处理分布式数据流。在文献[28]中，Bhaduri 和 Wolff 等人提出 PeDiT(P2P Decision Tree Induction Algorithem)算法，即 P2P 环境中建立决策树的算法。作者给出一种在分布式环境下以异步方式工作的交替算法，从而降低了通信负载，满足了扩展需求，而且算法对于数据的变化和网络节点的问题提供了自适应性。并且指出 PeDiT 算法也适用于无线传感器网络环境。

在文献[29]中，Chen 等人提出一种从分布的异构 Web 日志数据流中学习贝叶斯网络的综合方法。在他们的方法中，首先每个站点根据它的本地数据学习一个局部贝叶斯网络，然后标识出能够证实局部变量和非局部变量耦合的观测实例，并将这些观测实例的子集传输到中心站点。在中心站点，使用来自局部站点的数据学习另一个贝叶斯网络。最后，中心和局部贝叶斯网络合并成一个综合贝叶斯网络，为整个数据集建模。由于综合贝叶斯网络的参数可以根据多流的新数据不断更新，这个方法适合在线贝叶斯学习。同时，这个方法也适合挖掘资源约束环境下的分布式数据流。

2.4 聚类

聚类(Clustering)指对一个已给的数据对象集合，将其中相似的对象划分为一个或多个簇的过程。同一个簇中的元素彼此相似，而与其他簇中的元素相异。在分布式数据流情况下，需要以分布形式处理数据流，进行概要通信以及对数据点做全局聚类。Guha 等人在文献[30]中提出了一种基于 K 均值的数据流聚类方法，它可以使用较少的内存和时间对数据点进行聚类。其概要数据包括聚类中心和分配给聚类的数据点数。K 均值作为最基本的聚类算法，聚类结果产生一个近似真实聚类结果的常数因子。文献[30,31]表明，K 均值算法很容易扩展到分布式环境中。从本质上讲，分布式节点上的

聚类结果能够被合并，并且使用相同的近似因子可以得到全局聚类。与其他聚类方法相比，K 均值聚类的结果并不理想。能够产生比较理想聚类结果的其他算法是否能够扩展到分布式数据流环境，是一个值得研究的方向。

在文献[32]中，Januzaj 等人提出一个基于密度的分布式聚类算法 DBSCAN。从本质上讲，每个节点都能够建立一个局部的基于密度的聚类，然后与中心节点进行聚类概要通信。中心节点对所有来自其他节点的概要信息执行基于密度的聚类，找到全局聚类。然后，全局聚类结果被传回其他分布式节点，用来更新局部聚类。由于这个方法不能处理动态数据，在文献[33]中，作者表示基于密度的聚类算法可以以增量的方式执行。因此，提出一个分布式的、增量的 DBSCAN 算法。然而，它类似于 Januzaj 等人提出的分布式聚类算法，其结果质量也不能保证。

Beringer 和 Hullermeir 在文献[34]中研究了并行数据流的聚类方法。当数据流同步到达时，他们的目标是找出相关的数据流。作者使用指数加权滑动窗口来维持数据流，同时以增量的方式计算离散傅立叶变换。在这个变换空间中，间隔执行 K 均值聚类，属于同一类的数据流被认为是相关的。这个方法是集中式的。该方法也能够关联分布式数据流。此外，这个方法也可应用于在线数据流分析，对于扩展到分布式计算环境也是可能的。在这个方法中，傅立叶系数以增量方式交换，并且以局部聚合的方式汇总远程信息。而且，通过变换重要的系数，这个方法可能产生近似的聚类结果。

2.5 频繁项集挖掘

频繁项集挖掘的目标是发现数据集中满足最小支持度阈值的、至少出现 X 次的所有 X 项集。这些项集称作频繁项集，而最小支持度阈值由用户提供。例如，在购物数据分析方面，一个频繁二项集可以是{尿布，啤酒}，这意味着许多购买尿布的顾客也购买了啤酒。挖掘频繁项集需要较高的 CPU 和 I/O 负载，对于时刻增加和删除的动态数据集，从头挖掘代价昂贵。为了解决在动态数据集上挖掘频繁项集的问题，一些研究者已经提出了增量技术[35-40]。增量算法在本质上重用了以前挖掘的知识，并且合并更新的数据去计算新的频繁项集。但是环境可能是这样的：数据库分布在多个站点，每个站点都是以不同的速率更新，这就要求使用分布式异步频繁项集挖掘技术。

Otey 等人在文献[41]中提出一个频繁项集分布式增量算法，它能够以增量的方式在动态数据集中找到最大频繁项集。最大频繁项集的定义是：若频繁项集 X 的所有超集都是非频繁项集，则称 X 为最大频繁项集；将所有的最大频繁项集组成的集合称为完全最大频繁项集。此外，在分布式环境中挖掘频繁项集是可行的。分布式节点通过交换它们的局部最大频繁项集获得全局最大频繁项集的超集。然后，在所有节点之间交换超集来获得局部计数。在最后一轮通信中，通过执行修剪操作找到精确的全局最大频繁项集的集合。

为了能在常量空间挖掘所有频繁项集，Manku 和 Motwani 在文献[42]中提出了一种确定的 ε-近似算法（ε 为误差度）：Lossy Counting 算法。Lossy Counting 算法的基本思想是：在主存中维持数据流的一个样本集合，每当数据流到来一个数据项，若其值已经出现在样本集合中，则将相应的计数器加 1；否则，将新到的数据项以及该数据项此前在数据流中出现频率的上界（估计值）加入到样本集合中。数据流每到来 $1/\varepsilon$ 个数据项，Lossy Counting 算法对样本集合进行一次扫描，删除其中频率低于 εN（N 为当前数据流已经到来的数据项个数）的样本。这个算法用于集中模式下挖掘数据流的频繁项集，并不能直接应用到分布式环境中。Lossy Counting 算法结合文献[41]中提出的方法论后，使这个算法可以处理分布式数据流。

Manjhi 等人在文献[43]中扩展了 Lossy Counting 算法，使其能在分布式环境中挖掘数据流频繁项。核心问题在于：当合并来自多个节点的概要信息时，如何更好地组织贴近度，以使节点之间的通信负载最小。根据附带着精度梯度的概要从叶子节点传到根节点，并且进行增量合并，将过程表示为层次通信拓扑结构。他们通过研究两个可替代的、不兼容的优化目标，发现最优的精度梯度：(1)降低用于传递响应的中心节点负载和(2)降低最坏的情况下的通信链路负载。虽然这个方法仅仅是用来求解频繁项的，但将它扩展后用于挖掘频繁项集也是可行的。

2.6 离群点检测

离群点检测是数据挖掘的一个重要内容，在欺诈检测、网络鲁棒性分析和入侵检测等领域有着重要的应用。离群点检测的目标就是找到与数据集其余各点最不相同的数据点[44]，其目的是消除噪音或发现潜在的、有意义的知识。大多数离群点检测算法都是首先计算每一对点之间的距离，然后标识那些与所有其他点距离最大的点为离群点[45]。对于静态数据集，这是一个时间复杂度为 $O(n^2)$ 的算法。但这种方法很难扩展到分布式流数据集。在这种数据集中，点是以分布的方式到达节点的，当前节点不一定是计算节点，所以分析结果必须以累加的方式进行处理。这些限制使我们不可能采用纯粹以距离为基础的方法，必须转向启发式技术。许多离群点检测系统的核心问题就是实时确定异常或尽可能接近实时，同时这个问题也是其他数据流应用的核心问题。此外，很多时候，来自不同站点的数据使分布式流挖掘非常适合这个领域。在这一部分，我们回顾与分布式数据流挖掘密切相关的离群点检测内容。

在离群点检测中，各种特定的应用方法已经在相关文献中提到。文献[46]提出了一种分布式传感器网络偏差检测方法。它是专门为传感器网络环境开发的，应用于传感器异常检测方面。该方法能够进行传感器监控值的密度估计。当一个传感器的监控值同历史数据相比出现较大的波动时，这个传感器被标识为异常。在分布式计算环境中，这种计算一般发生在附近的传感器，只有当系统需要时，结果才报告给中央控制节点。

网络入侵检测是分布式离群点检测应用实例之一。目前趋势是要求一种分布式的互联网入侵检测方法。这些趋势之一是面向分布式的入侵和攻击，也就是说入侵和攻击是来自因特网上一些不同的主机。另外一个趋势是互联网异构特性的日益提高，同一子网的不同主机也许需要不同的安全要求。例如，在文献[47]中，为了实现多样化的安全需求，对于分布式防火墙提出了很多建议。此外，移动和无线计算为了避免出现一个集中位置，已创建了动态的、复杂的网络拓扑结构。有效地检测和预防这些攻击需要分布式节点协作。然而，一个节点本身只能收集当前围绕它的网络状态信息，这可能不

足以检测分布式攻击。如果节点共享网络审计数据、主机黑名单以及已知的网络攻击模型，每节点都可以构建一个更好的全局网络模型。

在文献[48]中，Otey 等人提出了一个面向分布式在线数据流的分布式离群点检测算法，用于处理分布式站点的网络数据。基于一系列属性依赖关系，他们定义了奇异评分函数，用来检测连续属性空间、分类属性空间和混合属性空间中的离群点。他们简洁地概要了必需的依赖信息，然后利用内存保持这些概要。为了能够在分布式数据流系统中找到精确的离群点，需要频繁交换内存中的概要信息。由于这些概要信息可能很大，在分布式数据流系统中，每个分布式计算节点仅仅与其他节点交换局部离群点。如果基于局部模型的分布式节点大多判断 A 点为离群点，则 A 点被认为是全局离群点。虽然这种方法仅能找到近似的离群点，但是作者表示这种启发式方法在真实数据中效果很好。同时，作者表明，为找到精确的离群点，需要交换大量的概要信息，这将导致过量的通信负载。为了能够在有限的通信负载情况下检测到更加精确的离群点，用概要信息中的决定性因素取代概要信息在内存中进行交换，是一种可行的方法。此外，他们指出大量的依赖信息造成了内存中的概要信息量过大，通过约减内存需求能够使这个算法应用到能源约束环境中。

Porras 等人在文献[49]中提出一个大型网络协同入侵检测的分布式方法 EMERALD。这种方法通过一个分级监控系统对网络进行分布式保护。其监控系统能够分析包括服务、域和企业级的网络数据。然而，EMERALD 并没有提供不同组织协同检测机制。Locasto 等人在文献[50]中，通过不同组织的协同增强了网络入侵检测水平。如果组织可以协同检测，那么每个成员都能够建立更好的全局网络行为模型和更精确的攻击检测模型（因为他们有更多的数据来估计模型参数），这些模型将能够进行更好的攻击识别和预测。作者通过交换 Bloom 过滤器来实现协同，每个 Bloom 过滤器都能对特定组织入侵检测系统所检测到的可疑主机 IP 地址列表以及可疑主机访问过的端口号进行编码。使用 Bloom 过滤器不但能够保证协同组织信息的可靠性，而且能够减少内存中的数据交换。这个方法的主要局限性就是交换的信息不足以识别分布式攻击。例如，当发起攻击的主机不在协同组织的监控列表中时，就不能识别这个攻击。但是，来自所有组织 IDS 所收集的合并审计数据足以检测这个攻击。要实现这样的系统，有两个问题需要解决：第一，每个组织都能够收集到无交集的属性集。协同组织必须事先对通用属性的使用达成一致。对于入侵检测通用标准的一些想法已经通过通用入侵检测框架（CIDF）[51]实现。第二个问题就是组织成员数据的隐私保护问题。使用 Bloom 过滤器去编码大量的属性集是不现实的。然而，目前隐私保护数据挖掘技术能够使组织进行协同检测，而不会危害其数据的隐私性[52-54]。

目前，有许多方法可用来检测拒绝服务攻击。基于前面提到的 CIDF，Lee 等人提出一个检测新型分布式攻击的方法，不但使各个节点可以共享他们监测到的分布式攻击信息，而且允许他们发布新的攻击模型。在文献[55]中，Yu 等人提出一个基于中间件来防止拒绝服务攻击的方法。他们利用虚拟专用操作环境（VPOE），通过运行中间件的设备实现协同。这些设备可以作为防火墙或是网络监控，而且他们的角色可以根据需求随时改变。设备包含如下组件：攻击检测组件、与其他设备协同的信号接收组件以及策略处理组件。

在无线自组网网络环境中，研究人员已经对网络入侵检测方面做了一些研究，例如文献[56,57]。在无线自组网中，节点通过无线媒介进行通信，网络的拓扑结构是动态的，节点必须协作，信息才能被路由到正确的目标终端。由于开放的无线通信媒介、动态拓扑和协作特性，无线自组网特别容易遭到网络入侵，而且分布式入侵检测也很困难。

为了防御入侵，张等人已经提出几种入侵检测的技术[56,58]。在他们提出的架构中，网络中的每个节点都参与检测和响应，每个节点都具有局部检测组件和协作检测组件。局部检测组件负责从本地审计数据中检测入侵。如果一个节点有足够的证据证明入侵正在发生，它就会对入侵发出响应。反之，它将通过协作检测组件启动一个全局入侵检测过程。节点之间的协作就是共享它们的检测状态，而不是它们的审计数据，因此要建立一个精确的全局网络入侵检测模型是很困难的。在这种情况下，全局水平的入侵可检测性是不可靠的。因此作者指出，由于远程节点可能被盗用，以及数据可能不可靠，他们仅仅使用局部数据。

在另一篇文献[57]，Huang 和 Lee 提出了另一种用于无线自组网入侵检测的方法，该方法主要是对网络结构本身的入侵攻击进行检测。这类入侵通常是损坏路由表、路由协议和拦截包，或者发起网络级别的拒绝服务。由于无线自组网通常使用电池运行，每个节点都不断运行自己的入侵检测系统是不符合成本效益的，特别是以威胁级别较低的水平运行。作者提出一个更有效的方法，即在簇中选择一个节点作为整个簇的监控节点（簇头）。作者假设，在传输范围内的每个节点都可以侦听到网络通信，同时其他的簇节点能够提供一些特征（由于簇头的传输范围和其他簇节点的传输范围不重叠，其他的簇节点会对不能访问簇头的簇部分节点进行统计），簇头负责分析簇中的数据包，检测入侵和发起响应。为使这种入侵检测方法更加有效，所选择的分簇算法应满足以下要求：第一，公平地、随机地选择一个节点作为簇头；第二，被盗用的节点既不能挤走当前簇头，也不能让簇头永久保持；第三，所选择的簇头不是被破坏的节点或恶意节点。这种方法使各个簇节点具有很好的工作分工，簇头运行入侵检测系统，其他节点负责收集数据并发送数据到簇头节点。然而，这个方法有局限性，在全局层面上并不是所有的入侵都是能够发现的，特别是考虑到检测系统的属性设置。例如，分析簇头通信内容并不能判断节点上服务漏洞造成的入侵。

3 资源约束环境下分布式数据流挖掘

目前，资源约束领域的分布式数据流挖掘已经成为研究热点。例如，在传感器网络领域，由于能量消耗的约束，过多的通信是不可取的。这样的环境下，潜在要求执行更多的计算，同时为了降低能耗，执行同样任务需要更少的通信。因此，这样的领域需要能够调谐计算和通信需求的数据挖掘算法[59,60]。

在网络入侵检测领域，一类相似的问题已经出现。为了增强系统可靠性和减少强加在主机处理环境上的约束，研究人员提出将卸载流量监控和入侵检测等计算应用到网卡上，即在网卡上实现流量监控和入侵检测技术。该领域的初步成

果传达了这种方法的可行性，但是这一代的网卡存在若干局限性（例如编程模型、缺乏浮点操作等），而这些问题在下一代网卡可能解决。

Kargupta 等人在文献[61]中提出 Mobimine 系统，这是一个基于 PDA(Personal Digital Assistant)的智能分析数据流系统，用于监控和分析股市实时数据流，同时将用户感兴趣的股票行为信息传给用户。所谓用户感兴趣的股票信息，是指那些对用户股票投资组合产生积极或消极影响的信息。而且，为了辅助用户分析，他们传送分类树给用户 PDA，分类树是由前文提到的频谱傅立叶方法表示的，这种表示方式非常适合网络带宽有限的环境。随着 PDA 设备计算能力的不断增强，新一代数据流分析挖掘系统越来越倾向于将更多的分析和挖掘功能从服务器端移植到 PDA 端，以降低整个系统的通讯代价。

文献[62]中提到的 VEDAS 系统，用于移动车辆的监控、分布式信息提取和数据流挖掘。VEDAS 不断获取和监测车载 PDA 实时产生的数据流，并实时进行模式提取，然后通过低带宽的无线网络传递给中央控制节点。该系统提供了诸如实时车辆状况监测、醉驾检测、驾驶特征和与商务车队管理相关的安全应用。在这种环境下的数据流挖掘等研究工作刚刚起步，相信在这个领域仍然有许多工作要做。

4 系统支持

一个完备的分布式数据流挖掘系统是非常复杂的，通常包括以下几部分：挖掘算法、通信子系统、资源管理器和调度管理等组件。一个成功的数据流挖掘系统必须适应动态的数据流，同时能够最佳地利用资源与组件的组合。在这部分，我们将简单概括资源感知型分布式数据流处理在系统支持方面的研究近况。

当系统处理连续数据流时，数据到达可能是突发性的，数据流速也可能是随着时间不断变化的。在如此的环境中，为了能够给予用户快速或实时的查询响应，系统必须能够在不损害性能的情况下轻松地处理突然到达的数据流。Babcock 等人在文献[63]中提到，一个算子调度策略的选择会对运行系统的内存利用率造成很大影响。特别是在数据流突发的情况下，错误的算子调度策略选择会很大程度上导致高的内存利用率和糟糕的系统性能。为了降低峰值负载时的内存利用率，作者提出了一种链式调度策略，这是一种自适应的、负载感知的查询算子调度策略。对于包含关系选择、关系投影和关系外键连接的单数据流查询来说，这个数据流算子调度策略在最小化运行时内存利用率方面是近优的策略。当峰值负载发生时，这个调度策略选择一个单位时间就能够处理和释放大量内存的算子路径（连续的算子集合）。事实上，这种方式不但使算子调度具有可选性，而且使系统具有较高的元组聚合处理速率。

前面提到的调度策略并不是面向分布式数据流处理的。而且，链式算子调度会对系统响应时间造成不利的影响，不适合应用于交互性能要求较高的数据流挖掘。为了挖掘数据流，需要一种既能满足响应时间又能支持内存感知的算子调度策略。此外，在分布式环境中，使用依赖算子调度一个数据流挖掘应用时，调度策略无需大量的状态信息通信。Ghoting 和 Parthasarathy 在文献[64]中提出一个自适应算子调度策略，它能够应用在分布式数据流挖掘环境中，而且能够保证响应时间和有限的内存利用率。用户可以通过调整应用到期望的交互水平来促进数据挖掘过程。作者通过响应时间的逐步退化来实现此调度策略，这种退化起始于对响应时间而言调度是最优的时刻。这种在响应时间方面的牺牲主要是用来优化内存利用率。如果做完一个起始的调度决策之后，系统状态的改变可能会迫使重新考虑算子调度。作者表示，对于局部状态信息的改变能否影响全局的算子调度，可由局部站点做出判定。因此，当系统状态变化较小时，局部站点可以独自处理。只有必要时，全局任务才被触发。

在文献[65]中，Plale 阐述了分布式环境下高效的时序连接处理问题。作者的目标是优化事务流的连接处理，使之更有效地判断同时发生的事务集合。为避免错过一些事务，窗口连接的大小不能够事先确定。作者提出可变窗口的概念，窗口的大小随着输入流的速率变化而变化。数据流入的速率能够很好地标识数据流中有多少事务已经过期。约减窗口的大小，也有助于降低内存利用率。而且，代替以先来先服务(FCFS)算法传送事务到查询引擎，作者提出将时间上最早的事务(EJF:Earliest Job First)先进行处理。这将有利于连接结果中部分事务的早期决策，同时提高了系统性能。

在文献[66]中，Chen 等人提出一个处理分布式数据流的中间件 GATES。这个中间件是建立在开放式网格服务架构上，针对网格环境下数据流处理的。它提供了一个高级接口，允许用户指定一个数据流处理算法作为一组流水阶段。GATES 的关键设计目标之一就是根据不断变化的环境提供自适应方案。为了支持自适应策略，GATES 根据不断改变的数据流环境，改变采样率、概要结构大小和挖掘算法中的一个或多个。例如，如果数据流速提高，系统通过降低采样率的方式获得实时响应。如果不改变采样率，系统将会面临不断增长的队列长度，最后导致糟糕的系统性能。为了能够支持自适应性，系统设计人员需要提供带有参数的中间件，这些参数使用户可以根据数据流环境的变化实时对系统进行调节。GATES 建立了一个简单的性能模型，用来预测参数如何变化才有助于分布式环境下的性能自适应。

Chi 等人在文献[67]中提出一个挖掘多数据流的负载脱落策略。作者认为，从流中读取数据并提取特征值的过程计算复杂度较高，常常成为系统的瓶颈，问题的核心在于如何使用有限的计算实现多数据流的特征提取。他们根据历史流中数据项的效用来决定是否放弃当前流中的数据项。如果选择不提取当前项的特征值，则占用有限内存的马尔可夫链根据历史数据来预测当前项的特征值。虽然作者提出的方法是集中的，但其中的负载决策方案很容易扩展到分布式环境中。

5 结论和展望

本文讨论了当前分布式数据流挖掘的最新进展。具体来说，讨论了概要提取、Skyline 查询、分类、聚类、频繁项集挖掘和离群点检测算法。此外，简要论述了分布式数据流挖掘的相关应用和系统支持。

虽然国内外的研究人员在分布式数据流挖掘方面已取得了较大的进展，但要构建实用的、高精度的大型分布式数据流挖掘系统，仍有一些亟待解决的问题，下面分别予以阐述。

1)需要挖掘的分布式数据源有可能跨越多个组织，这些

组织之中还可能存在异构的计算资源。此外，分布式数据源常常被多个执行不同挖掘任务的分析软件访问。到目前为止，已经提出的各种分布式数据流挖掘系统都没有考虑挖掘任务的多样性和计算资源的有限性。为了使分布式环境下的系统便于执行和部署，考虑组织隐私保护的、即插即用的系统设计是十分必要的。具有层次服务的架构将有利于开发分布式数据流挖掘的快速应用。此外，这些系统需要同已有的数据网格和知识网格架构集成，而且研究者需要为此集成设计中间件。

2)分布式环境是数据流挖掘应用增长较快的领域。但由于分布式环境的复杂度和安全问题，传统集中式数据流挖掘隐私保护方法无法直接应用于分布式环境中。而且分布式数据流挖掘系统可能会处理跨越多个组织的流数据源。而这些不同组织的流数据源均有独立的安全机制，且在物理上是分布式的或异构的。在这些流数据源上进行数据挖掘，原先需要通过不同认证才能获得授权许可的机密数据，在实施分布式数据流挖掘系统后，仅单一认证即可访问获取。根据木桶原理，整体系统的安全性降至较低的水平。因此，如何在分布式环境下进行带有隐私保护的数据流挖掘，是一个值得研究的方向。

3)下一代数据挖掘的计算系统很有可能是建立在使用宽带互联的现成处理器上。为了能够在此系统上获得高性能，需要重新设计数据流算法。例如，当前多核处理器已经成为主流，正如文献[68]所示，内存墙(memory-wall)问题会对数据挖掘算法造成不利的影响。而且对于多核架构，这个问题会更加恶化。因此，为了在多核架构上获取更高的性能，需要重新设计局部站点的数据流挖掘算法。与此类似，伴随着网络技术的创新，需要研究考虑到高性能互联技术(例如 Infiniband)的算法设计。

4)从分布式数据流挖掘系统的实现框架角度考虑，一个有效的分布式框架能够在很大程度上提高数据流挖掘的效率。MapReduce 是 Google 开发的分布式框架，用于大规模数据集的并行运算，也是云计算的核心技术。它简化了分布式编程模式，适用于处理大量数据的分布式运算，同时可以用于解决问题的程序开发模型。将数据流挖掘算法与 MapReduce 框架相结合，设计更加有效的分布式数据流处理算法，构造可靠的、高性能的、可扩展的分布式数据流挖掘系统，是未来可行的研究方向。

5)在很多应用实例中，需要分布式数据流挖掘的环境都是资源约束型的。因此，需要研究适应特殊执行环境的数据流挖掘方法。各种各样的权衡必须在此类环境中进行评估，例如能源和通信、通信和冗余计算等。所以，需要设计具有可调节计算和通信需求功能的数据流挖掘算法。尽管在这个领域已经有一些初步尝试，但是对各种各样权衡的系统评估还没有完成，即使对单一的应用领域也没实现。进一步展望未来，基于特定的方法，为大量的应用领域开发更加抽象的接口集合，将是一个很值得研究的方向。

6)分布式数据流挖掘新的应用即将出现。例如，射频识别技术(RFID)通过允许自动捕捉和识别，预计会显著提高商务流程的效率。在不远的将来，预计 RFID 芯片会被嵌入到多种设备中，被捕捉到的数据可能普遍存在。因此，分布式数据流集新的应用将会出现，需要研究新的数据挖掘方法。

结束语 在过去几年里，相关文献中已经提出了很多数据流挖掘算法。虽然这些算法能够在集中的环境下运行，但它们中的许多不能够运行在分布式环境下，也不能轻易扩展。为了能够在分布式环境下获得精确或近似的结果，系统需要交换大量的状态信息。为了方便分布式数据流挖掘算法的设计，取代集中式解决方案，我们从一开始就需要有分布式的思维形式。应该设计从分布式、增量环境中高效获得概要信息的算法，然后提出使用这些概要信息的特定方法。这样的设计策略将会有利于分布式数据流挖掘算法的设计。

参 考 文 献

[1] Babcock B, Babu S, Datar M, et al. Models and issues in data stream systems[C]// Proceedings of the Symposium on Principles of Database Systems (PODS). 2002:1-16

[2] Bulut A, Singh A. SWAT: Hierarchical stream summarization in large networks[C]// Proceedings of the International Conference on Data Engineering (ZCDE). 2003:72-76

[3] Papadimitriou S, Sun J, Faloutsos C. Streaming pattern discovery in multiple time series[C]// Proceedings of the International Conference on Very Large Data Bases (VLDB). 2005:697-708

[4] Chiky R, Hébrail G. Summarizing Distributed Data Streams for Storage in Data Warehouses[C]// Proceedings of the 10th International Conference on Data Warehousing and Knowledge Discovery. Turin, Italy: Springer-Verlag, 2008:65-74

[5] Babcock B, Olston C. Distributed top-k monitoring [C]// Proceedings of the International Conference on Management of Data (SIGMOD). 2003:28-39

[6] Borzsonyi S, Kossmann D, Stocker K. The skyline operator[C]// Proceedings of the 17th International Conference on Data Engineering. Washington: IEEE Computer Society, 2001:421-430

[7] Tao Y F, Papadias D. Maintaining sliding window skylines on data streams[J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(3):377-391

[8] Chomieki J, Godfrey P, Gryz J, et al. Skyline with presorting[C]// Proceedings of the 19th International Conference on Data Engineering. Washington: IEEE Computer Society, 2003:717-719

[9] Papadias D, Tao Y F, Fu G, et al. Progressive skyline computation in database systems[J]. ACM Transactions on Database Systems, 2005, 30(1):41-82

[10] Tan K, Eng P, Ooi B. Efficient progressive skyline computation [C]// Proceedings of the 27th VLDB Conference. Roma, Italy, 2001:301- 310

[11] Kossmann D, Ramsak F, Rost S. Shooting stars in the sky: an online algorithm for skyline queries[C]// Proceedings of the 28th VLDB Conference. Hong Kong, China, 2002:275-286

[12] Beckmann N, Kriegel H-P, Schneider R, et al. The R*-tree: An Efficient and Robust Access Method for Points and Rectangles [C]// Proc. of the ACM SIGMOD Conference. Atlantic City, NJ, 1990:322-331

[13] Guttman A. R-Tree: A dynamic index structure for spatial searching[C]// Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 1984:47-57

[14] Tan Y, Papadias D. Maintaining sliding window skylines on data streams [J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(3):377-391

[15] Lin X, Yuan Y, Wang W, et al. Stabbing the sky: efficient skyline computation over sliding windows[C]// Proceedings of the 21st International Conference on Data Engineering. 2005:502-513

[16] Balke W, Untzer U, Zheg J. Efficient distributed skylining for Web information systems[C]// International Conference on Extending Database Technology. Heraklion, Grece, 2004:256-273

[17] Wu P. Parallelizing skyline queries for sealable distribution parallelizing skyline queries for scalable distribution[C]// International Conference on Extending Database Technology. Munich, Germany, 2006:112-130

[18] Hose K. Processing skyline queries in P2P systems[C]// VLDB 2005 PhD Workshop. Trondheim, Norway, 2005:36-40

[19] Wang S, Ooi B C, Tung A K H, et al. Efficient skyline query processing on peer-to-peer networks[C]// Proc. of ICDE. 2007: 1126-1135

[20] Huang Z, Jensen C S, Lu H. Skyline queries against mobile lightweight devices in manets[C]// Proceedings of the 22nd International Conference on Data Engineering. 2006

[21] Yoon S, Shahabi C. Distributed Spatial Skyline Query Processing in Wireless Sensor Networks[C]// International Workshop on Sensor Webs, Databases, and Mining in Networked Sensing Systems (SWDMNSS) in conjunction with International Conference on Networked Sensing Systems (INSS). 2009

[22] Sun S, Huang Z, et al. Efficient monitoring of skyline queries over distributed data streams[J]. Knowledge and Information Systems, 2009, 25(3):575-606

[23] 孙圣力,李金玖,朱扬勇. 高效处理分布式数据流上 skyline 持续查询算[J]. 软件学报, 2009, 20(7):1839-1853

[24] 王爱冬,张涛,阳国贵. 分布式数据流上的 Skyline 计算[J]. 计算机工程与应用, 2008, 44(1):151-154

[25] Hulten G, Domingos P. Mining high speed data streams[C]// Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD). 2000:71-80

[26] Jin R, Agrawal G. Efficient decision tree construction on streaming data[C]// Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD). 2003: 571-576

[27] Kargupta H, Park B. A fourier spectrum based approach to represent decision trees for mining data streams in mobile environments[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(3):216- 229

[28] Bhaduri K, Wolff R, et al. Distributed Decision-tree Induction in Peer-to-Peer Systems[J]. Stat. Anal. Data Min. 2008, 1(2): 85-103

[29] Chen R, Sivakumar D, Kargupta H. An approach to online Bayesian network learning from multiple data streams[C]// Proceedings of the International Conference on Principles of Data Mining and Knowledge Discovery. 2001:31-45

[30] Guha S, Mishra N, Motwani R, et al. Clustering data streams[C]// Proceedings of the Symposium on Foundations of Computer Science (FOCS). 2000:359-366

[31] Ghoting A, Parthasarathy S. Facilitating interactive distributed data stream processing and mining[C]// Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS). 2004

[32] Januzaj E, Kriegel H, Pfeifle M. DBDC: Density-based distributed clustering[C]// Proceedings of the International Conference on Extending Data Base Technology (EDBT). 2004:231-244

[33] Ester M, Kriegel H, Sander J, et al. Incremental clustering for mining in a data warehousing environment[C]// Proceedings of the International Conference on Very Large Data Bases (VLDB). 1998:323-333

[34] Beringer J, Hullermeier E. Online clustering of parallel data streams[J]. Data and Knowledge Engineering, 2005:180-204

[35] Cheung D, Han J, Ng V, et al. Maintenance of discovered association rules in large databases: An incremental updating technique[C]// Proceedings of the International Conference on Data Engineering (ICDE). 1996:106-114

[36] Cheung D, Lee S, Kao B. A general incremental technique for maintaining discovered association rules[C]// Proceedings of the International Conference on Database Systems for Advanced Applications. 1997:185-194

[37] Ganti V, Gehrke J E, Ramakrishnan R. DEMON: Mining and Monitoring Evolving Data[J]. IEEE Transactions on Knowledge and Data Engineering, 2001, 13(1):50-63

[38] Lee S, Cheung D. Maintenance of discovered association rules: When to update[C]// Proceedings of the Workshop on Research Issues in Data Mining and Knowledge Discovery. 1997:51-58

[39] Thomas S, Bodagala S, Alsabti K, et al. An efficient algorithm for the incremental updation of association rules in large databases[C]// Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD). 1997: 263-266

[40] Veloso A, Meira W Jr, De Carvalho M B, et al. Mining frequent itemsets in evolving databases[C]// Proceedings of the SIAM International Conference on Data Mining. 2002:31-41

[41] Otey M, Wang C, Parthasarathy S, et al. Mining frequent itemsets in distributed and dynamic databases[C]// Proceedings of the International Conference on Data Mining (ICDM). 2003: 617-620

[42] Manku G, Motwani R. Approximate frequency counts over data streams[C]// Proceedings of the International Conference on Very Large Data Bases (VLDB). 2002:346-357

[43] Manjhi A, Shkapenyuk V, Dhamdhere K, et al. Finding (recently) frequent items in distributed data streams[C]// Proceedings of the International Conference on Data Engineering (ICDE). 2005:767-778

[44] Barnett V, Lewis T. Outliers in Statistical Data[J]. John Wiley and Sons, 1994, 12(1):175-176

[45] Knorr E, Ng R T. Algorithms for mining distance-based outliers in large datasets[C]// Proceedings of the International Conference on Very Large Data Bases (VLDB). 1998:392-403

[46] Palpanas T, Papadopoulos D, Kalogeraki V, et al. Distributed deviation detection in sensor networks[J]. SIGMOD Record, 2003:77-82

[47] Ioannidis S, Keromytis A D, Bellovin S M, et al. Implementing a distributed firewall[C]// ACM Conference on Computer and Communications Security. 2000:190-199

[48] Otey M, Ghoting A, Parthasarathy S. Fast distributed outlier detection in mixed attribute data sets[J]. Data Mining and Knowledge Discovery Journal, 2006:203-228

(下转第 36 页)

[4] 闵栋,刘东明,徐迎阳. 基于 Mashup 的移动互联网业务架构研究[J]. 数据通信,2009(2)

[5] 高永兵,吴纪磊,胡文江,等. 基于 Web 服务的 Mashup 应用的研究与实现[J]. 计算机技术与发展,2010,20(6)

[6] 黄家乾,吴升. Mashup 技术在 web 地图中的应用[C]// 中国地理信息产业论坛论文集. 武汉:中国地理信息系统协会,2009

[7] Wang A B, Zhang J, Jiang W R. Useful Resources Integration Based on Google Maps[C]// Proc. of 2009 4th International Conference on Computer Science & Education. 2009:1044

[8] Qian Z, Wei XL. The Research and Implementation of a RESTful Map Mashup Service[C]// Proc. of 2010 Second International Conference on Communication Systems, Networks and Applications. 2010:401

[9] Xu K, Zhang X Q, Song M N, et al. Mobile Mashup: Architecture, Challenges and Suggestions[C]// Proc. of Management and Service Science(MASS) International Conference. 2009:1

[10] Jin L, Song M N, Song JD. Mobile Mashup Architecture Solution, Direction and Proposal[C]// Proc. of 2010 IEEE 2nd Symposium on Web Society (SWS). 2010:698

[11] http://www.google.com/webelements

[12] 王辉,高成英,刘宁. 服务器端 Mashup 开发平台的设计与实现[J]. 计算机工程,2010,36(10)

[13] http://code.google.com/intl/zh-CN/apis/maps/documentation/javascript/services.html#Geocoding

[14] 汪军,符涛. 下一代网络中的业务融合模式[J]. 中兴通讯技术,2008,14(4)

[15] 廖康. 电信固网运营商转型中信息超市的研究及其建设[J]. 电子学报,2007,35(4)

[16] 秦灵伶,王文东,贾霞,等. Mashup 技术及其发展趋势[J]. 电信科学,2009,25(9)

[17] 闵栋,刘东明,徐迎阳. 面向移动互联网的 Mashup 聚合业务研究[J]. 现代电信科技,2009,39(3)

[18] 杨勇,贾霞,董振江. 电信业务能力开放技术标准[J]. 中兴通讯技术,2009,15(2)

[19] http://open.189works.com/app/developDocument

[20] 赵治. 步入多元化的电信 2.0 时代[J]. 电信技术,2009(5)

(上接第 8 页)

[49] Porras P, Neumann P. EMERALD: Event monitoring enabling responses to anomalous live disturbances[C]// Proceedings of the National Information Systems Security Conference. 1997:1-16

[50] Locasto M, Parekh J, Stolfo S, et al. Collaborative distributed intrusion detection[R]. Columbia University, 2004

[51] Lee W, Nimbalkar R A, Yee K K, et al. Stolfo. A data mining and CIDF-based approach for detecting novel and distributed intrusions[J]. Lecture Notes in Computer Science, 2000, 1907:49-65

[52] Kargupta H, Datta S, Wang Qi, et al. On the privacy preserving properties of random data perturbation techniques[C]// Proceedings of the International Conference on Data Mining (ICDM). 2003:99-106

[53] Kantarcioglu M, Clifton C. Privacy-preserving distributed mining of association rules on horizontally partitioned data[C]// Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. 2002:1026-1037

[54] Lindell Y, Pinkas B. Privacy preserving data mining [J]. ACM SIGMOD Record, 2000, 29(2):439-450

[55] Yu Wei, Xuan Dong, Zhao Wei. Middleware-based approach for preventing distributed deny of service attacks[C]// Proc. of IEEE Military Communications (MILCOM). 2002:1124-1129

[56] Zhang Yong-guang, Lee W, Huang Yi-aa. Intrusion detection techniques for mobile wireless networks[J]. Wireless Networking, 2003, 9(5):545-556

[57] Huang Yi-an, Lee W. A cooperative intrusion detection system for ad hoc networks[C]// Proceedings of the 1st ACM Workshop on Security of ad Aoc and Sensor Networks. 2003:135-147

[58] Zhang Yong-guang, Lee W. Intrusion detection in wireless ad-hoc networks[J]. In Mobile Computing and Networking, 2000:275-283

[59] Kargupta H. Distributed data mining for sensor networks[C]// Tutorial presented at ECML/PKDD. 2004:77-82

[60] Palpanas T, Papadopoulos D, Kalogeraki V, et al. Distributed deviation detection in sensor networks[C]// SIGMOD Record. 2003:77-82

[61] Kargupta H, Park B, Pittie S, et al. MobiMine: Monitoring the stock market from a PDA[J]. SIGKDD Explorations, 2002, 3(2):37-46

[62] Kargupta H, Bhargava R, Liu K, et al. Vedas: A mobile and distributed data stream mining system for real-time vehicle monitoring[C]// Proceedings of the SIAM International Conference on Data Mining (SDM). 2004:18-26

[63] Babcock B, Babu S, Datar M, et al. Chain: Operator scheduling for memory minimization in data stream systems[C]// Proceedings of the International Conference on Management of Data (SIGMOD). 2003:253-264

[64] Ghoting A, Parthasarathy S. Facilitating interactive distributed data stream processing and mining[C]// Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS). 2004

[65] Plale B. Learning run time knowledge about event rates to improve memory utilization in wide area stream filtering[C]// Proceedings of the International Symposium on High Performance Distributed Computing (HPDC). 2002:171-178

[66] Chen L, Reddy K, Agrawal G. GATES: A grid-based middleware for processing distributed data streams[C]// Proceedings of the International Symposium on High Performance Distributed Computing (HPDC). 2004:270-277

[67] Chi Y, Yu P, Wang H, et al. Loadstar: A load shedding scheme for classifying data streams[C]// Proceedings of the SIAM International Conference on Data Mining (SDM). 2005:342-361

[68] Ghosting A, Buehrer G, Parthasarathy S, et al. A characterization of data mining algorithms on a modern processor[C]// Proceedings of the ACM SIGMOD Workshop on Data Management on New Hardware. 2005:1-5