一种基于 DBMS 的无监督异常检测算法及其应用*)

钟 勇[1,2] 林冬梅[1] 秦小麟[2]

(佛山科学技术学院信息与教育技术中心 佛山 528000)[1]

(南京航空航天大学信息科学与技术学院 南京 210016)[2]

摘 要 传统的基于身份认证和存取控制的数据库安全机制存在一定的局限性,如无法防止 SQL 注入、合法用户权限滥用等非法行为,而现存的入侵检测研究多集中在网络和操作系统,由此提出一个基于 DBMS 的无监督异常检测算法。首先定义了数据库查询的表示方法及其相似度计算方法,其次给出了包括查询聚类、标记和检测三阶段的异常检测算法,最后给出了算法在合成数据中的聚类结果及其在真实数据中检测 SQL 注入的应用,并讨论了利用数据库索引的扩展算法。

关键词 聚类算法,数据库安全,异常检测

An Algorithm of Unsupervised Anomaly Detection Based on DBMS and its Application

ZHONG Yong[1,2] LIN Dong-Mei[1] QIN Xiao-Lin[2]

(Information and Educational Technology Center, Foshan University, Foshan 528000)[1]

(Information Science and Technology Institute, Nanjing University of Aeronautics and Astronautics, Nanjing 210016)[2]

Abstract There are limitations on the traditional user identification and access control of database security mechanism, such as in preventing the illegal actions of SQL injection, misusing authorization. However, most of existed intrusion detection researches focus on network or operation system, so the paper presents an algorithm of unsupervised anomaly detection based on DBMS. Firstly, the paper defines the expression of database queries and similarity computation between queries. Then an anomaly detection algorithm that includes three phases: clustering, labeling and detecting is given out. Finally, an experiment result on a synthetic data set and a result on a real data set for detecting SQL injection are reported, and the modified algorithm based on index also is discussed at the end of the paper.

Keywords Clustering algorithm, Database security, Anomaly detection

1 引言

1.1 相关的工作

传统的数据库安全机制以身份认证和存取控制为重点,身份认证和存取控制主要着眼于对外部用户的身份和权限约束的检查以确定用户或其操作的合法性,是以预防为中心的被动安全机制,无法满足日益增长的对数据库安全的需要,特别是计算机网络化的发展,使数据库面临着前所未有的安全困境,一些来自网络的攻击,攻击者往往能窃取到合法的身份或权限,如利用密码嗅探(password sniffing)攻击者可能获得合法的用户账号和密码,利用会话骑劫(session hijacking)的攻击者可能伪装成合法的用户,利用 SQL 注入[1](SQL injection)入侵者可能执行非法的查询语句。数据库应该具有更加主动、积极的安全机制才能有效地防止网络互联给数据库带来的层次不穷的攻击。因此近年来,对主动型数据库安全机制的研究受到广泛的重视。入侵检测通过对运行系统的状态和活动进行检测,分析出非授权的访问和恶意行为,从而发现入侵行为和企图,是一种重要的主动安全防范技术。但以往对入侵检测的研究大多基于网络和操作系统,对数据库则较少涉及。而对数据库来说,仅仅依靠工作在文件和系统命令级的底层操作系统和网络入侵检测系统无法保证检测的效率和精度,如 SQL 注入常常利用数据库应用程序的漏洞,这种入侵是操作系统和网络入侵检测系统所难以检测的。而数据库中的数据具有自己的结构和语义,数据库用户有自己的独特行为,通过数据库入侵检测可以弥补操作系统和网络入侵检测的不足,提高检测的准确度和有效性。

在相关的方法中,Christina 等[2]通过挖掘经常同时被查询的数据项作为用户轮廓来识别异常查询,该方法假定用户在使用数据库系统时有一定的一致性,如果该假定不成立或非一致性阈值的设置不正确,该方法有较大的误检率(False Positive)。Lee 等[3]使用合法事务 SQL 语句的正则表达式("指印")来代表正常用户的行为,该方法易产生过多的"指印"且只对某些特定的入侵如 SQL 注入才有效,我们在以前的研究中[4]提出从用户的 SQL 语句中挖掘频繁项集并通过计算用户查询与频繁项集的偏离度来确定查询的异常度,由于偏离度计算公式存有一定的误差,因此该方法也有较大的误检率。Yi 等[5]通过数据项中的依赖关系来检测恶意事务,该方法只能应用于恶意事务的检测中,应用范围较窄。由于数据库结构的复杂性,数据库入侵检测技术面临着更多的研究难点,技术上还处在研究阶段,完善的实用系统尚未见到。

1.2 本文的工作

本文提出了一个基于 DBMS 的无监督异常检测算法,该

*)航空科学基金(02F52033)、江苏省高技术项目(BG2004-005)资助。钟 勇 副教授,博士研究生,当前主要研究方向为数据库安全、网络安全;秦小麟 博士生导师,教授,当前主要研究方向为安全数据库、空间数据库、时空数据库、信息安全等。

算法主要具有以下特点：

(1)该算法通过将数据库查询描述为查询结果集中元组标识符集和属性集的集合，提出查询的相似度计算及其聚类方法，由于用户进行查询的主要目地在于得到数据库中的数据，查询结果集能更精确地描述用户的行为，因此，与现存方法相比，该方法具有更高的精确度。

(2)该算法是一种无监督的异常检测方法，由于获取纯净数据和有标签数据的难度，传统的异常检测算法经常受到限制[6]。无监督算法输入无标记数据(unlabeled data)，输出数据中的异常数据，不需要或几乎不需要数据的先验知识和领域知识，在许多场合具有更多的优势。另外，无监督方法的处理结果常可作为其它异常检测方法的数据预处理步骤。

(3)利用数据库索引地址作为元组标识符，该方法能够应用在数据库查询执行前进行异常检测，防止恶意查询的执行造成无法挽回的破坏。

2 相似度定义

首先定义单个关系上的查询相似度，假定查询基于如下关系 R：

定义 1(基本定义) 让 R 是定义在属性 $A=\{a_1,\cdots,a_n\}$ 上的关系，属性集 A 的属性分别定义在域 $D_1,\cdots,D_n$ 上。R 的任意元组 $t:t\in D_1\times\cdots\times D_n$，将关系 R 上的关系代数选择和投影操作分别表示为：选择操作 $\sigma_F(R)=\{t\mid t\in R\wedge F(t)=$'真'$\}$，其中 F 是逻辑表达式，取逻辑值'真'或'假'。逻辑表达式 F 的基本形式为 $X_1\Theta Y_1[\phi X_2\Theta Y_2]\cdots$，其中 Θ 表示比较运算符，$\Theta\in\{=,\neq,>,<,\leqslant,\geqslant\}$。$X_i$ 和 Y_i 是属性名或常量或简单函数，ϕ 表示逻辑运算符 $\phi\in\{\neg,\wedge,\vee\}$，[]表示任选项，…表示上述格式可以重复下去；投影操作 $\Pi_{SA}(R)=\{t[A]\mid t\in R\}$，其中 SA 为 R 中的属性列。

在不失一般性的情况下，假定数据库关系中的每一元组都由一独特的标识属性 ID 唯一标识，属性 ID 可以是元组的物理地址、逻辑地址、主键或其它能唯一识别元组的属性，另外为方便起见，假定属性 ID 隐藏(不出现在属性集中)于查询的所有结果集中(利用其值可唯一确定一元组)，使用表 1 的用户关系示例，其中属性集 $A=$｛帐户，密码，性别，系，年龄｝，关系集中有 6 个元组。

表 1 用户关系示例

ID	帐户	密码	性别	系	年龄
1	Zyq830	A123	男	数学	27
2	TianBel	B346	女	计算机	36
3	Jimmy	H459	女	数学	48
4	Ljr3000	GC37	男	数学	45
5	Liu6021	NN14	男	计算机	36
6	Rodger117	A563	男	化学	27

定义 2(查询) 关系 R 上的查询 Q 是三元组$\langle qid,A_q,F\rangle$，其中 qid 是查询 Q 的唯一性标识符，$A_q\subseteq A$ 是 R 的属性列，F 是查询的逻辑表达式，直观起见，本文将查询表示成 $Q=(qid,F\rightarrow A_q)$，称 $F\rightarrow A_q$ 是 Q 的查询内容，F 是 Q 的前件，A_q 是 Q 的后件。

在定义 2 中，并未考虑更新、插入查询和聚集查询以及一些包含函数在其后件中的查询，但对这些查询作相应改变可以变换成定义 2 的格式，如对更新查询，可将查询分为读部分和写部分；对插入查询，可将查询作为写部分且将插入的值作为前件，插入属性作为后件；对后件中包含的函数可将其展开，例如对 SQL 语句表示的查询"select $\alpha^*\ a_i+\beta^*\ a_j$ from T where F"，可得到该查询的内容为 $F\rightarrow\{a_i,a_j\}$。

定义 3(查询结果集) 对关系 R 上的查询 $Q=(qid,F\rightarrow A_q)$，让函数 $RS(qid)$ 返回 Q 的结果集，为方便表示，让函数 $attr(qid)$ 返回结果集中的属性集合(即查询的后件)，$attr(qid)=A_q$；让函数 $tid(qid)$ 返回结果集中元组标识(ID 属性)的集合。

查询结果集是关系 R 上的子集，如对表 1 的查询 $Q=$｛101，性别='女'$\rightarrow$｛帐户，性别｝｝，得到表 2 的查询结果集 $RS(101)$，则 $attr(101)=$｛帐户，性别｝，$tid(101)=\{2,3\}$。

表 2 查询结果示例

ID	帐户	性别
2	TianBel	女
3	Jimmy	女

定义 4(查询摘要，Query Summary，QS) 对关系 R 上的查询 $Q=(qid,F\rightarrow A_q)$，定义 Q 在关系 R 上的查询摘要为：$sum(Q)=\{qid,tid(qid),attr(qid)\}$。

如上述查询的查询摘要为：$sum(Q)=\{101,\{2,3\},$｛帐户，性别｝｝。

定义 5(查询聚类) 给定训练集中所有查询的标识集合 QID，定义查询聚类 $C=\{qid\mid qid\in QID\}$，查询聚类是标识集合 QID 的子集，有 $C\subseteq QID$。让函数 $attr(C)$ 返回 C 中查询所包含的属性集，即 $attr(C)=\{a\mid a\in attr(qid)\wedge qid\in C\}$；让函数 $tid(C)$ 返回 C 中各查询所包含的元组标识集合，即 $tid(C)=\{a\mid a\in tid(qid)\wedge qid\in C\}$。

定义 6(聚类支持度) 对给定的查询聚类 C，对任意属性 $A_i\in A$，定义 A_i 在聚类 C 上的支持度：$sup(A_i,C)=|\{qid\mid A_i\in attr(qid)\wedge qid\in C\}|$；对任意元组标识 t，定义 t 在聚类 C 上的支持度：$sup(t,C)=|\{qid\mid t\in tid(qid)\wedge qid\in C\}|$。

定义 7(聚类摘要，Cluster Summary，CS) 给定任意查询聚类 C，定义 C 的元组标识简化内容 $tidCont(C)=\{(t,sup(t,C))\mid t\in tid(C)\}$；定义 C 的属性简化内容 $attrCont(C)=\{(a,sup(a,C))\mid a\in attr(C)\}$，那么，定义 C 的聚类摘要 $sum(C)=\{C,tidCont(C),attrCont(C)\}$。

查询摘要和聚类摘要是本文聚类算法的主要数据结构，除了这两个结构，不需要关于查询和聚类的其它知识。

定义 8(查询相似度) 对关系 R 上的查询 $Q_1=(qid_1,F_1\rightarrow A_1)$ 和 $Q_2=(qid_2,F_2\rightarrow A_2)$，定义 Q_1 和 Q_2 之间的相似度为：

$$sim(Q_1,Q_2)=\frac{(2*|tid(qid_1)\cap tid(qid_2)|)}{|tid(qid_1)|+tid(qid_2)|}*\frac{(2*|attr(qid_1)\cap attr(qid_2)|)}{|attr(qid_1)|+|attr(qid_2)|}\qquad(1)$$

查询之间的相似度是查询摘要中元组标识集 Dice 系数和属性集 Dice 系数[7]的乘积，Dice 系数常用在信息检索中对集合相似度的测量，查询摘要是元组标识集和属性集的集合，当两个查询之间的元组标识符集和属性集之一的 Dice 系数为零时，则两个查询是完全不同的查询，其相似度为 0，故公式(1)使用两个 Dice 系数的乘积而不是和。由于 Dice 系数的结果在[0,1]之间，故有 $0\leqslant sim(Q_1,Q_2)\leqslant 1$，易证 $sim(Q_1,Q_2)=sim(Q_2,Q_1)$。

定义 9(聚类相似度) 对关系 R 上的聚类 C_1 和 C_2，定

义 C_1 和 C_2 之间的相似度为：

$$sim(C_1,C_2)=\frac{\sum_{p\in tid(C_1)\cap tid(C_2)}(sup(p,C_1)+sup(p,C_2))}{\sum_{p\in tid(C_1)}sup(p,C_1)+\sum_{p\in tid(C_2)}sup(p,C_2)}*\frac{\sum_{a\in attr(C_1)\cap attr(C_2)}(sup(a,C_1)+sup(a,C_2))}{\sum_{a\in attr(C_1)}sup(a,C_1)+\sum_{a\in attr(C_2)}sup(a,C_2)} \tag{2}$$

可以证明，当 C_1 和 C_2 都仅包含单个查询时，公式(2)即是公式(1)，当 C_1 只包含单个查询 Q 时，由公式(2)可得到查询 $Q=(qid,F\rightarrow A_q)$ 与聚类 C 的相似度：

$$sim(Q,C)=\frac{\sum_{p\in tid(Q)\cap tid(C)}(sup(p,C)+1}{\sum_{p\in tid(C)}sup(p,C)+|tid(qid)|}*\frac{\sum_{a\in attr(Q)\cap attr(C)}(sup(a,C)+1)}{\sum_{a\in attr(C)}sup(a,C)+|attr(qid)|} \tag{3}$$

同样易证性质 $0\leqslant sim(C_1,C_2)\leqslant 1$ 和 $sim(C_1,C_2)=sim(C_2,C_1)$。

上述定义建立在单关系查询上，对多关系查询，可以使用查询在各个关系上的相似度的平均值作为多关系查询的相似度。

定义 10（多关系查询相似度） 给定多关系查询 $Q_1=(qid_1,F_1\rightarrow A_1)$ 和 $Q_2=(qid_2,F_2\rightarrow A_2)$ 及其所在的关系集 R^* 和 S^*，定义 Q_1 和 Q_2 之间的相似度为：

$$sim(Q_1,Q_2)=\frac{\sum_{R\in R^*\cap S^*}sim_R(Q_1,Q_2)}{|R^*\bigcup S^*|} \tag{4}$$

其中，$sim_R(Q_1,Q_2)$ 是 Q_1 和 Q_2 在关系 R 上利用公式(1)得到的结果，对多关系聚类之间和聚类与查询之间的相似度也可类似于本公式使用平均值来计算。

3 异常检测算法

使用类似于文[6]的异常检测算法，简单起见，只考虑单关系查询，算法包括以下 3 部分算法：训练算法、标记算法和检测算法。

训练算法输入训练集 Qu 中表示成摘要形式的查询和阈值 δ，输出所有的查询聚类，算法使用一种单联接(single-linkage)聚类算法的变体，当第一个查询摘要读入时，查询摘要插入到新的聚类摘要中，当下一个查询摘要读入时，算法计算该查询摘要与现存聚类的最大相似度，如果最大相似度大于阈值 δ，则将该查询摘要插入最大相似度所在的聚类，否则创建新的聚类并将该查询摘要插入，重复该过程直到读入所有查询，最后输出所有聚类。

算法 1 训练算法

输入：查询训练集 Qu，阈值 δ

输出：k 个查询聚类

假设条件：假定 Qu 中所有输入查询已经过预处理，表示成查询摘要(QS)形式。

步骤 1：初始化聚类集 C 为空集。

步骤 2：从训练集 Qu 中依次读入查询 q 的摘要(QS)，如果 C 为空集，则生成新的聚类摘要 CS 并将查询 q 的摘要并入 CS 中，否则在聚类集 C 中找到与 q 有最大相似度的聚类 s。

步骤 3：如果聚类 s 与查询 q 的相似度大于等于阈值 δ，则将 q 的摘要并入 s 的摘要中，否则生成新的聚类摘要 CS 并将查询 q 的摘要并入 CS。

步骤 4：重复步骤 2、3 直到读入所有 Qu 中所有的查询，然后输出聚类集 C。

标记聚类算法假设正常数据占训练数据的绝大部分(>98%)，由此，包含正常数据的聚类很可能大于包含异常数据的聚类，将百分之 P 的最大数量的聚类标记为正常，其它聚类标记为异常。

算法 2 标记算法

输入：聚类集 C，百分比 P

输出：已标记的聚类集 C

步骤 1：对 C 中聚类包含查询数的多少从大到小排序；

步骤 2：将 C 中聚类的前百分之 P 的聚类标记为正常，其余标记为异常。

在训练阶段，算法已能够识别正常和异常查询，如果需要将训练阶段的结果作为实时检测的轮廓，则可以使用如下的异常检测算法，找出已标记好的聚类集中与实时查询具有最大相似度的聚类，将该聚类的标记作为实时查询的标记，具体算法如下：

算法 3 检测算法

条件：已标记好的聚类集 C

输入：实时查询 q

输出：q 的标记

算法步骤：

步骤 1：在 C 中查找与实时查询 q 相似度值最大的聚类 c；

步骤 2：输出 c 的标记作为 q 的标记。

在检测阶段，由于要对实时查询进行检测，对性能要求较高。检测算法的复杂度主要在于找到聚类集 C 中与实时查询 q 中最大相似度的聚类，假定 C 中的聚类数为 n，算法需要执行 n 次 q 与聚类相似度的计算。

4 实验与应用分析

本节使用合成数据来分析聚类结果，使用实际数据来分析算法应用。

4.1 合成数据

IBM 数据合成器 QUEST(IBM Quest Market-Basket Synthetic Data Generator)[8] 主要用于产生数据挖掘的事务数据，我们将查询的元组集合和属性集合分别看作是元组和属性上的事务，改写 QUEST 使之同时产生两个事务，一个事务代表元组标识集，一个事务代表属性集，从而合成查询摘要。产生元组数 500、属性数 20 的关系，以及产生平均元组标识数为 10、平均标识模式长度为 6、平均属性数 8、平均属性模式长度为 4 的查询 2000 条，在不同阈值下得到的聚类数如表 3 所示。

表 3 聚类结果集

聚类大小 \ 阈值(δ)	0.1	0.2	0.3	0.4	0.5	0.6
1	8	25	95	208	399	678
2	1	3	15	28	28	23
3	0	3	5	8	14	7
(3, 5]	3	3	12	14	17	13
(5, 10]	1	10	19	23	20	28
(10, 30]	7	24	29	31	34	38
≥30	18	17	20	19	17	7
聚类总数	38	85	195	331	529	794

当阈值增大时，要求同一聚类的查询具有更高的相似度，造成聚类数目增加，特别是小聚类数目的增加，从表 3 可以看到这一趋势，这也显示阈值 δ 对检测的精度有着重要的影响。

4.2 真实数据

佛山科技学院的学生在上网前必须输入帐号和密码，将用户每次输入的用户名和密码值看作是查询结果中的元组标识值，用户属性和密码属性则作为查询结果的两个属性。简单起见，选取 1 月内 10 个用户的 495 条登录数据，在不同阈值下得到表 4 的聚类结果。

表 4 对登录数据的聚类结果

阈值(δ)	聚类数	聚类大小（括号中数字表示该大小聚类的数量）
0.4	23	83,71,66,63,51,32,29,28,27,26,4,2(3),1(9)
0.5	28	83,71,66,61,51,29,27,26(2),25,4,3,2(7),1(9)
≥0.6	42	80,70,64,60,50,26(2),24(2),23,3,2(14),1(17)

对表 4 中最大的 10 个聚类分析发现，这些聚类中主要是正确的用户名和密码的登录数据，当阈值较小时也包括部分易拼错（多次输入错误）而导致的错误用户名或错误密码的数据，但不包括两者都错的数据，而其它更小的聚类包括错误用户名或密码或两者都错的数据，这些可看作是异常查询聚类。

SQL 注入是一种入侵者利用字符串技术伪造恶意 SQL 语句来欺骗应用程序的入侵手段，SQL 注入往往是由于应用程序的设计疏忽或不完善等问题导致的。例如现今流行的在互联网上使用的三层 B/S 体系结构中，用户不直接查询数据库而是通过应用层来代理用户的请求，如使用 SQL SERVER 数据库的 T-SQL 事务语句和 C# 伪码的代码片段[9]。

```
[SqlCommandMethod(CommandType. Text,"SELECT * FROM Users WHERE Username=@username AND Password=@Passwd")]
public static DataSet GetAnnouncements(SqlConnection connection, int moduleId)
{…functions}
```

在 T-SQL 中，字符串表示成'***'，分号是 SQL 语句的结束符，符号--表示单行注释。如果应用程序没有为变量 @username 和@Password 执行适当的输入验证，恶意用户可能进行 SQL 注入攻击，如用户在@username 变量中输入"admin'--"，查询将执行"Select * fromusers where username='admin'--"；输入"'or 1=1"，查询将执行"Select * from users where username=''or 1=1"。这些语句都可以使用户绕过正常的密码检查。甚至用户可以在@username 变量中输入"'; drop table user--"，查询将执行"Select * from users where username ='';drop table user--"，导致用户表删除。输入代码验证如符号替换、输入限定、长度限制等方法常用来对付 SQL 语句注入，但这些方法也存在限制和问题，如文[1]所示。

表 5 SQL 注入语句在不同阈值下的聚类结果

阈值(δ)	SQL 注入语句所在聚类的大小值	检测率	误检率
0.2	M_1(28),M_2(33),M_3(1),M_4(1),M_5(84)	40%	0
0.4	M_1(1),M_2(33),M_3(1),M_4(1),M_5(1)	80%	20%
0.5	M_1(1),M_2(1),M_3(1),M_4(1),M_5(1)	100%	28.57%

在表 4 中除去异常查询增加下列 SQL 注入语句，再重新聚类的结果如表 5 所示。从下表可以看到，随着阈值增大，检测率提高，当阈值 $\delta \geq 0.50$ 时，所有的 SQL 注入语句都可以检测到，说明本文的算法检测 SQL 注入是有效的。但当阈值增大时，误检率也随之提高，检测率和误检率在实际应用中是一对需要仔细权衡（tradeoff）的因素。

```
(M1)select * from account where username='Anson-cao'.
(M2)select * from account where password='8278560'.
(M3)select * from account where username=''.
(M4)select * from account where username='error username' and password='error password'.
(M5)select * from account where username=''or 1=1.
```

其中'caihonghui' 和 '90bnrd'分别是合法的用户名和密码。

5 基于数据库索引的扩展算法

上述算法必须在查询执行后的结果集中进行分析，在许多情况下，如查询执行的时间较长或查询具有不可逆行为，如能在查询执行之前进行检测，对异常查询或提出执行警告或拒绝执行，可有效提高系统性能和减少损失。在现在的大多数关系数据库中，都存在一定的索引机制。索引表的每一项是由一个属性值和一个指针（即存储位置）构成的二元组(k, p)，k 是对应记录的属性值，p 是该记录的地址。如果将 p 直接看作是元组的标识符，那么通过索引，可以得到元组的标识符集，从而可应用本文的相似度计算公式。

对查询 $Q=(qid, F\rightarrow A_q)$，首先将逻辑表达式 F 转换成合取范式 $F=C_1 \wedge C_2 \cdots \wedge C_n$，其中任意 C_i 是任意析取式 $C_i=E_1 \vee E_2 \cdots \vee E_m$，对每一个 E_j，通过索引可得到该表达式的地址集合（元组标识符集），再通过集合操作则可得到 Q 的地址集合（元组标识符集）。如对表 1 的 SQL 查询语句 q="select 帐户，密码 from 用户表 where 性别='男'or（系='数学'and 性别='女'）"，转换成合取范式的查询为"（性别='男'∨系='数学'）∧（性别='男'∨性别='女'）→{帐户，密码}"，从表 6 的系和性别索引，可得到该查询的标识符集（地址集）为：$tid(q)=(\{5,1,3,4\}\cup\{1,3,4\})\cap(\{5,1,3,4\}\cup\{6,2\})=\{1,3,4,5\}$。

表 6 系和性别索引

系	地址	性别	地址
化学	6	女	6
计算机	2	女	2
计算机	5	男	5
数学	1	男	1
数学	3	男	3
数学	4	男	4

由于篇幅的原因，本文不再讨论基于索引的扩展定义和算法，使用基于索引的方法虽然能对实时查询进行执行前检测，但需要对每一个逻辑表达式中的相关属性建立索引，对大数据量关系需要占用大量的存储空间，而且如果 E_j 中包括多个属性需要进行索引间的连接操作，则对检测速度也有较大影响。但如果索引与关系分开存放，该方法易于并行操作，可同时执行索引上的检测和实际的查询，则该方法具有较高的效率。

结束语 本文提出了一个基于 DBMS 的无监督异常检测算法，该算法通过将数据库查询描述为查询结果集中元组标识符集和属性集的集合，提出查询的相似度计算及其聚类方法，从而得到本文的三阶段异常检测算法，该算法是一个无监督的异常检测算法，与现存方法相比，该方法具有更高的精确度，且能应用在实时查询执行前进行检测。由于传统的数据库安全机制越来越无法满足现代数据库的安全需要，作为最后一道安全防线的入侵检测系统起着至关重要的作用，基于 DBMS 的异常检测算法的研究有着重要的理论和现实意义。

参 考 文 献

1 Anley C. Advanced SQL Injection In SQL Server Applications. http://www. nextgenss. com/papers/advanced_sql_injection. pdf, 2002
2 Christina Y C, Michael G, Karl L. DEMIDS: A Misuse Detection System for Database Systems. In: Proc. of the Third Annual IFIP TC-11 WG 11.5 Working Conference on Integrity and Internal Control in Information Systems. Amsterdam, Netherlands, 1999. 158～178
3 Lee S Y, Low W L, Wong P R. Learning Fingerprints for A Database Intrusion Detection System. In: ESORICS 2002. Lecture Notes in Computer Science, No 2502 Springer-Verlag, 2002. 264～280
4 Zhong Y, Qin X L. Research on Algorithm of User Query Frequent Itemsets Mining. In: Proc. of Third International Conference on Machine Learning and Cybernetics. Shanehai, China, Aug. 2004. 1671～1676
5 Yi H, Brajendra P. A data mining approach for database intrusion detection. In: Proc. of the 2004 ACM Symposium on Applied Computing. Nicosia, Cyprus, March 2004. 711～716
6 Portnoy L, Eskin E, Solfo S. Intrusion detection with unlabeled data using clustering. In: Proc. of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001). Philadelphia, PA: November, 2001
7 Moses S C. Similarity estimation techniques from rounding algorithms. In: Proc. of the thirty-fourth annual ACM symposium on Theory of computing. Montreal, Quebec, Canada, 2002. 380～388
8 IBM Quest Market-Basket Synthetic Data Generator. http://www. cs. indiana. edu/～cgiannel/assoc_gen. html
9 O'Neill M. Web服务安全技术与原理. 冉晓旻,郭文伟译. 北京: 清华大学出版社, 2003. 168～169

(上接第 122 页)

接万用表测电压的方法实现,另外 MoteLab 所支持的网络规模较小,扩展性不强。

4.2 Kansei

俄亥俄州立大学开发的 Kansei 平台是面向多种应用的针对无线传感器网络的测试平台。Kansei 平台在设计上充分考虑了对大规模应用环境的支持以及对各种应用背景的通用化和可扩展性的要求,为无线传感器网络测试平台的搭建提供了启发性的思路。

从结构上划分,Kansei 平台由静止网络、便携网络和移动网络三部分组成。静止网络是由 210 个节点组成的矩形规则阵列。每个节点分为 XSM 节点和 Stargate 单板计算机两部分。XSM 节点使用 916MHz 微波通信(更新的版本使用了 IEEE802.15.4 协议通信),是网络测试和控制的对象。Stargate 拥有独立的系统使用 IEEE802.11b 的协议通讯,与以太网相连并可以通过一个 51 脚的连接器对 XSM 进行访问和控制。移动网络由 5 个机器小车组成,行驶于静止网络节点上面铺设的玻璃板上。移动网络节点可以用于收集反馈信号并向静止网络实时注入数据,从而配合静止网络完成测试。便携网络中节点数量不定,除了进行数据存储、压缩、传输和时间同步的管理以外,根据不同试验的需求选择搭配不同的传感器,用于在实际的应用环境中进行数据的感应和采集。

静止网络和移动网络共同构成了 Kansei 系统中的测试通用平台部分,部署在实验室环境中。便携网络则根据测试应用类型选择相应的传感器,部署到实际的测试环境中进行数据采集。便携网络所采集的数据通过以太网发送至 Kansei 的软件平台 Director 上。在 Director 中先对数据建立基于物理参数特性的模型,再通过概率插值等方法将数据扩展到静止网络和移动网络中,从而建立了一个混合模拟的通用测试平台(每个节点都可以模拟扩充为多个节点),以供研究者进行测试。

Kansei 平台的三种网络结构为搭建真实反映大规模应用环境的测试平台提供了可行性的思路。首先,利用部署、回收便利的便携网络可以在实际的环境条件下进行数据采集,从而更加真实地反映了数据的空间特性和应用特点,而且便携网络的设计也提高了 Kansei 平台面对多种应用的扩展性和灵活性。其次,通过实际节点与理论模拟相结合的混合模拟方法,有效地解决了测试平台网络节点规模不够大的问题。最后,移动网络的设计使得 Kansei 平台可以对移动无线传感器网络应用进行测试评估,网络结构更加丰富灵活。不过目前 Kansei 平台还处于开发过程中,如系统访问控制等功能并没有完全实现,混合模拟方法的效果也有待进一步验证。

总结和展望 随着无线传感器网络应用研究的不断深入,人们越来越意识到通过实际传感器节点建立真实网络平台进行协议和算法测试的重要性。无线传感器网络平台测试涉及到网络体系结构设计、节点及网络状态测量、网络重编程以及对多种应用特性支持等多种问题。本文对其中的三个关键技术:测试评估技术、监视控制技术以及平台搭建技术分别进行了讨论,并总结了其目前发展现状,为如何建立测试平台提供了指导性的介绍。

目前,国内外关于这方面的研究处于初始阶段,在今后的工作中,如何形成系统化标准化的测试评估体系以及如何使测试平台无论在网络规模及应用范围上具有更好的扩展性,将成为无线传感器网络平台测试技术研究的重点。

参 考 文 献

1 李建中,李金宝,石胜飞. 传感器网络及其数据管理的概念、问题与进展. 软件学报,2003,14(10):1717～1727. http://www. jos. org. cn/ 1000-9825/14/1717. htm
2 Cerpa A, Busek N, Estrin D. Scale: A tool for Simple Connectivity Assessment in Lossy Environments. Center for Embedded Networked Sensing, University of California, Los Angeles: [Tech. Rep. CENS Technical Report 0021]. September 2003
3 Girod L, Elson J, Cerpa A, et al. Emstar: a software environment for developing and deploying wireless sensor networks. In: Proceedings of the 2004 USENIX Technical Conference, Boston, MA, 2004
4 Wang Y, Martonosi M, Peh L S. A new scheme on link quality prediction and its applications to metric-based routing. In: Proceedings of the 3rd international conference on Embedded, 2005
5 Hull B, Jamieson K, Balakrishnan H. Mitigating congestion in wireless sensor networks. ACM SenSys, 2004
6 Shnayder V, Hempstead M, Chen B, et al. Simulating the power consumption of large-scale sensor network applications. on Embedded Networked Sensor Systems (SenSys' 04)
7 Landsiedel O, Wehrle K, Gotz S. Accurate Prediction of Power Consumption in Sensor Networks. In: Proc. 2nd IEEE Workshop on Embedded Networked Sensors, May, 2005
8 Zhao Y J, Govindan R, Estrin D. Residual Energy Scan for Monitoring Sensor Networks. WCNC, 2002
9 Ritter H, Schiller J, Voigt T, et al. Experimental Evaluation of Lifetime Bounds for Wireless Sensor Networks. EWSN, 2005
10 Levis P, Culler D E. Mate: a tiny virtual machine for sensor networks. ASPLOS, 2002. 85～95
11 Boulis A, Han C C, Srivastava M B. Design and Implementation of a Framework for Efficient and Programmable Sensor Networks. ACM MobiSys, San Francisco, 2003
12 Liu T, Martonosi M. Impala: a middleware system for managing autonomic, parallel sensor systems. In: Proceedings of the Ninrh ACM SIGPLAN Sympusium on Principles and Pracrice of Parallel Programming, ACM Press, June 2003
13 Cheong E, Liebman J, Liu J, et al. TinyGALS: A Programming Model for Event-Driven Embedded Systems. In: Proceedings of the 18th Annual ACM Symposium on Applied Computing (SAC'03), March 2003
14 Gummadi R, Gnawali O, Govindan R. Macro-programming Wireless Sensor Networks using Kairos. In: Proceedings of DCOSS'05, International Conference on Distributed Computing in Sensor Networks, June 2005
15 Levis P, Gay D, Culler D. Active Sensor Networks. www. tinyos. net/papers/
16 Werner-Allen G, Swieskowski P, Welsh M. MoteLab: A Wireless Sensor Network Testbed. ISPN, 2005
17 Ohio State University. Kansei: Sensor Testbed for At-Scale Experiments. Poster, 2nd International TinyOS Technology Exchange, Berkeley, CA, Feb. 2005