数据挖掘技术报告

域名服务器信息挖掘

学 校：北京理工大学

指导老师：汤世平 老师

小组成员：王恒怿 2120151038

苏思悦 2120151031

郑越 2120151072

目录

摘要 0
第一章 背景 1
1.1 域名服务器 1
1.2 域名路径解析 2
1.3 网络异常 3
第二章 相关研究 5
2.1 相关概念 5
2.2 相关算法分析 7
2.2.1 研究现状 7
2.2.2 本文的研究基础 8
2.3 本文的贡献 9
第三章 基于 Apriori 的频繁连续时间片段选择算法 11
3.1 算法概述 11
3.2 实现步骤 11
3.3 算法要点 13
第四章 基于 Apriori 的频繁域名解析路径分析算法 15
4.1 算法概述 15
4.2 实现步骤 15
第五章 结果分析 17
5.1 基于 Apriori 的频繁连续时间片段选择算法实现 17
5.1.1 数据源及实现环境 17
5.1.2 频繁序列挖掘结果举例 17
5.2 基于 Apriori 的频繁域名解析路径分析算法 19
第六章 展望 21
第七章 总结 22
参考文献 23

摘要

本文主要对七天的会话集进行预处理，实现了频繁域名路径解析和域名服务器流量预测。在域名路径解析过程中，我们使用 java 编程实现了 Apriori 算法。在服务器流量预测过程中，我们实现了论文中[2]提出的算法，使用 Apriori 算法查找频繁段，并在此基础上，创新使用了后缀数组算法处理数据，很大的提升了算法的速度。查找到的频繁段用于聚类域名服务器，方便服务器的部署优化，并可预测流量攻击。

关键字： 域名解析 流量预测 Apriori 算法 后缀数组

第一章 背景

在信息时代，网络的生命在于其安全性和可靠性。网络是人类发展史来最重要的发明，提高了科技和人类社会的发展。计算机网络最重要的方面是它向用户所提供的信息服务及其所拥有的信息资源，给用户带来方便。

1.1 域名服务器

网络服务器是网络环境下能为网络用户提供集中计算、信息发表及数据管理等服务的专用计算机。根据不同的计算能力，服务器又分为工作组级服务器、部门级服务器和企业级服务器。服务器操作系统是指运行在服务器硬件上的操作系统。服务器操作系统需要管理和充分利用服务器硬件的计算能力并提供给服务器硬件上的软件使用。

域名服务器(Domain Name Server简称DNS),保存了一张域名(domain name)和与之相对应的IP地址（IP address）的表，以解析消息的域名；是一种组织成域层次结构的计算机和网络服务命名系统，它用于TCP/IP网络，它所提供的服务是用来将主机名和域名转换为IP地址的工作。域名是Internet上某台计算机或者计算机组的名称，用于在数据传输时标识计算机的电子方位（有时也指地理位置)。域名是由一串用点分隔的名字组成的，通常包含组织名，而且始终包括两到三个字母的后缀，以此来指明组织的类型或者该域所在的国家或者地区。

域名系统作为一个层次结构和分布式数据库，包含各种类型的数据，包括主机名和域名。DNS数据库中的名称形成一个分层树状结构称为域命名空间。域名包含单个标签分隔点，例如：im.qq.com。完全限定的域名 (FQDN) 唯一地标识在 DNS 分层树中的主机的位置,通过指定的路径中点分隔从根引用的主机的名称列表。

把域名翻译成IP地址的软件称为域名系统，即DNS。它是一种管理名字的方法。这种方法是：分不同的组来负责各子系统的名字。系统中的每一层叫做一

个域，每个域用一个点分开。所谓域名服务器（即 Domain Name Server，简称 Name Server）实际上就是装有域名系统的主机。它是一种能够实现名字解析（name resolution）的分层结构数据库。

1.2 域名路径解析

当 DNS 客户机需要查询程序中使用的名称时，它会查询本地的 DNS 服务器来解析该名称。客户机发送的每条查询消息都包括三条消息，以指定服务器应答问题。

©指定的 DNS 域名，表示为完全合格的域名（FQDN）。

©指定的查询类型，可根据类型指定资源记录，或作为查询操作的专门类型。

©DNS 域名的指定类别。

DNS 查询以各种不同的方式进行解析。客户机有时也可通过使用从以前查询获得的缓存信息就地应答查询。DNS 服务器可使用其自身的资源记录信息缓存来应答查询，也可代表请求客户机来查询或联系其他 DNS 服务器，以完全解析该名称，并随后将应答返回至客户机，这个过程称为递归。

另外，客户机自己也可尝试联系其他的 DNS 服务器来解析名称。如果客户机这么做，它会使用基于服务器应答的独立和附加的查询，该过程称作迭代，即 DNS 服务器之间的交互查询就是迭代查询。具体过程如图 1.2-1 所示：

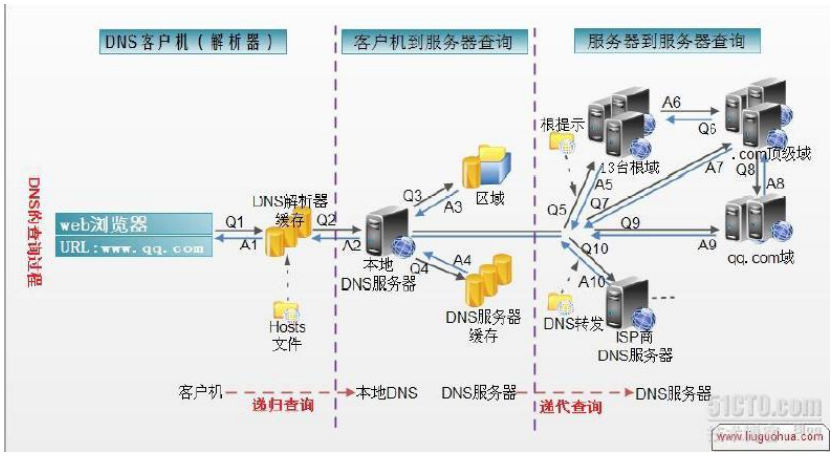

图 1.2-1 DNS 服务器路径解析

通过对域名解析路径进行数据分析，挖掘出域名解析路径中的频繁模式，用于优化服务器部署，避免出现某一服务器过“过忙”的现象。对于域名解析路径的挖掘我们采用基于 Apriori 的频繁域名解析路径分析算法，该算法会在第四章中进行详细讲解。

优化服务器部署，合理地选择路径，让轻负载时非最佳的路径，在重负载时有较多的分流；重负载时增加分层连接分流的连接数目；增大通道的贷款；增加信息速率；适当增加缓冲区等。减少用户对资源需求的办法：拒绝某些服务请求；要求用户减少负载量；合理配备用户对资源的使用，如使用预约、轮询、假如优先级等。减少用户对资源的需求的办法，其实质是降低服务水平和质量，或合理进行服务。

1.3 网络异常

网络安全是一个关系国家安全和主权、社会的稳定、民族文化的继承和发扬的重要问题。其重要性，正随着全球信息化步伐的加快而变得越来越重要。网络安全是指网络系统的硬件、软件及其系统中的数据受到保护，不受偶然的或者恶意的原因而遭到破坏、更改、泄露，系统连续可靠正常地运行，网络服务不中断。从其本质上来讲就是网络上的信息安全。从广义来说，凡是涉及到网络上信息的保密性、完整性、可用性、真实性和可控性的相关技术和理论都是网络安全的研究领域。 从网络运行和管理者角度说，他们希望对本地网络信息的访问、读写等操作受到保护和控制，避免出现“陷门”、病毒、非法存取、拒绝服务和网络资源非法占用和非法控制等威胁，制止和防御网络黑客的攻击。

被动攻击虽然难以检测，但可采取措施有效地预防，而要有效地防止攻击是十分困难的，开销太大，抗击主动攻击的主要技术手段是检测，以及从攻击造成的破坏中及时地恢复。检测同时还具有某种威慑效应，在一定程度上也能起到防止攻击的作用。具体措施包括自动审计、入侵检测和完整性恢复等。 本文通过

对域名服务器进行流量监测，对数据进行分析处理，统计得出每一服务器流量访问的大致趋势，据已有的研究发现，在非异常的情况下，用户对于某一域名的访问都是有一定的时间规律；从而来对每一时间段内的流量访问量进行分析，采用基于 Apriori 的频繁连续时间片段选择算法，该方法会在第三章节中详细讲述，预测出该服务器下一时间段的状态，以此来推测是否可能发生异常。

第二章 相关研究

随着信息技术的快速发展，信息数据、信息访问量以指数方式飞速增长，人类步入了数据时代。无论是在商业活动、社会、科学、工程领域，还是在教育、医学等领域，每天都有惊人数量的数据汇入计算机网络中。为了方便、快捷、体系地处理爆炸式增长的可用数据，从海量数据中发现有价值的信息，进而把信息转化为有组织、易于理解的知识，数据挖掘应运而生。

本章 2.1 小节主要介绍了数据挖掘的相关概念，以及解决 DNS 流量相关问题的专业术语概念。为了更快捷的进行流量预测、准确的挖掘访问路径的频繁项集，并作出相应的攻击预测，我们仔细学习研究了现有的几个经典算法，并在其之上做了改进实现，因此本章 2.2 小节主要介绍了这些现有的基础算法，并对之进行分析。最后，基于现有的研究基础和理论基础，本章第 2.3 小节系统介绍了本文的主要贡献——基于 Apriori 的 DNS 访问流量和路径频繁性分析技术。

2.1 相关概念

所谓数据挖掘，是指从大量的数据中挖掘有趣模式和知识的过程。如图 2.1-1 所示，其发现知识的过程由数据清理、数据集成、数据选择、数据变换、数据挖掘、模式评估几个步骤迭代而成。

首先对多个数据源的数据进行清理，以消除噪声和不一致的数据，然后按照一定的规则方法将清理过后的数据进行系统的集成，然后在合并后的数据中进行数据选择，以筛选出与所分析任务相关的数据，然后

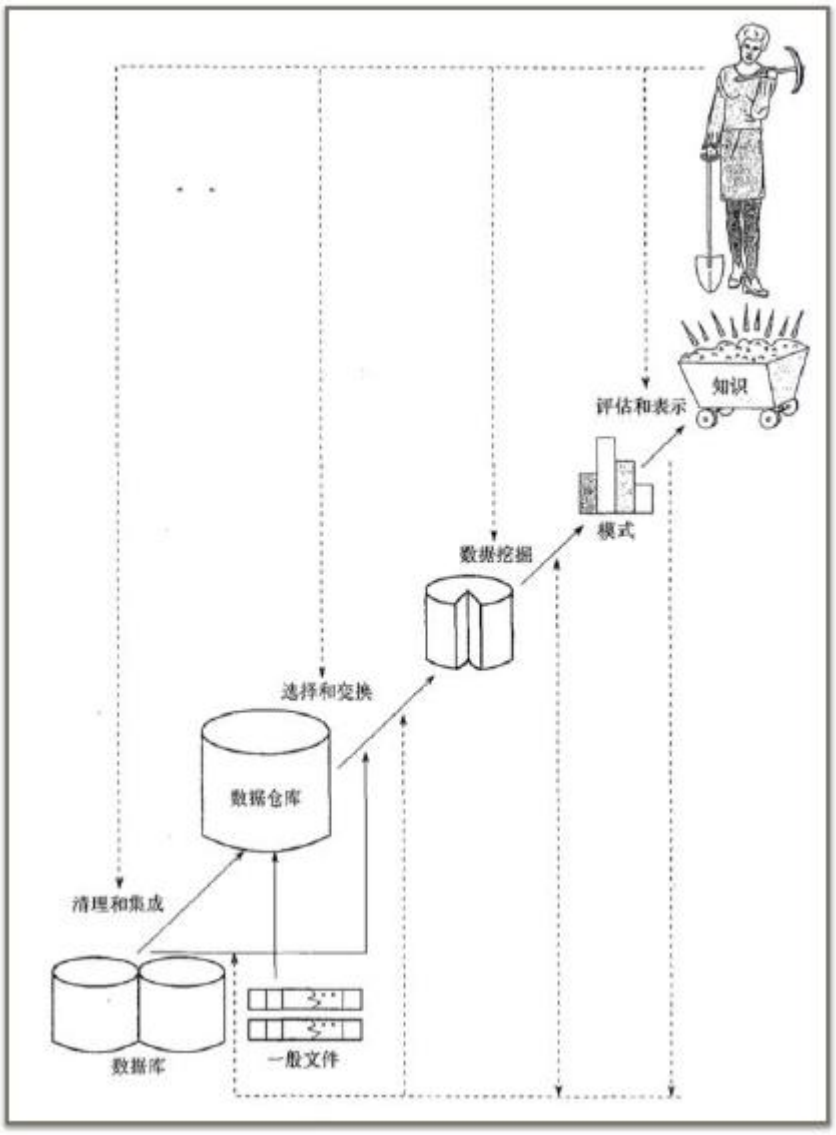

图 2.1 数据挖掘过程步骤图

进行一系列的数据变换操作，通过数据汇总、聚集，把数据变换和统一成适于挖掘的形式，然后通过数据挖掘，使用智能的方法提取出数据模式、关联规则等，然后进行模式的评估，根据特定的兴趣度度量，识别代表知识的真正的有趣模式，最后，用可视化和知识表示技术，向用户提供挖掘的知识表示。

数据挖掘的数据可以来源于各种各样的数据源，数据形式囊括数据库数据、数据仓库数据、事务数据、及其他各种类型的数据。基于一定的算法技术等，可以对这些数据挖掘频繁模式、关联和相关性，为后续的应用、决策提供信息支持。

频繁模式是在数据中频繁出现的模式，其形式呈现多样化，包括频繁项集、频繁子序列和频繁子结构。其中，频繁项集一般是指频繁的在事务数据集中一起出现的商品的集合；频繁子序列是一个频繁的序列模式；而频繁子结构则可能涉及不同的结构形式，一旦某个子结构频繁出现，就构成了一个频繁子结构。对这些频繁模式进行挖掘，进而可以发现有趣的关联和相关性，然后通过设置支持度阈值和置信度阈值，筛选出关联性符合要求的关联规则，然后通过进一步分析，发现相关联的属性－值之间的有趣的统计相关性。由此可见，从数据集中快速准确地挖掘出频繁模式在数据挖掘过程中起着举足轻重的作用。

如本文第一章所述，对 DNS 访问情况的流量分析、服务器部署、攻击预测等具有其必要性和迫切性，是当前的一大研究热点。这就必须要求发现 DNS 访问会话纪录中的规律、特点、关联关系等，因此也就必须挖掘出域名访问流量的时间变化规律和路径的频繁模式。在 DNS 访问过程中，会话信息数据按照一定的结构形式组织存在。这些数据数量庞大，关联关系复杂，难于观察分析，并且在每时每刻不断更新，用传统的人工分析方式，不仅会产生巨大的人力、物力开销，其分析的实时性和准确性也无法得到有效的保障。因此，必须采用数据挖掘的方法方式，挖掘流量随时间的变化关系，以及路径访问的频繁项集，进而基于这些特征，进行服务器部署、攻击预测。

2.2 相关算法分析

2.2.1 研究现状

随着 DNS 访问快速性、安全行、可靠性等问题变成当前的热点问题，越来越多的人关注于快速，便捷实现 DNS 的流量分析、异常检测、攻击预测等。如 Ji 等就提出了一种基于 k-means 的聚类算法以实现对 IP 地址和域名的时间行为的聚类【1】。该方法将域名分成四个聚类。这个方法并不是简单的比较不同域名的访问流量，而是对一系列的生成变量进行聚类，例如 DNS 的请求总数，两个 DNS

访问请求之间的平均时间间隔等等。尽管该方法可以产生有用的结果，但是它需要预先知道集群的数量 k，而预先知道这个值并不容易。

再比如 Wang 等提出了一种检测互联网上全国性大规模攻击的数学方法【2】。该方法用一个平均协方差表示正常访问时两个省份之间进行 DNS 查询的时间戳，然后构建协方差矩阵，纪录当前的时间戳，当两个发生明显差异时，认为极有可能发生网络异常甚至是攻击。然而这种方法仅仅适用于全国性的攻击，同时，仅仅能检测到针对特定域名的攻击检测，并不能具有较高的通用型。 类似的研究结果还有很多，然而各有其适用性和局限性所在。本文在研究了前人的研究基础之上，改进并实现了一种基于 Apriori 的 DNS 访问流量和路径频繁性分析技术。经测试，在已给的测试集上，能够表现出良好的分析、预测特性。

2.2.2 本文的研究基础

（1）Apriori 算法

Apriori 算法是 Agrawal 和 R.Srikant 于 1994 年提出的，为布尔关联规则挖掘频繁项集的原创性算法。该算法使用频繁项集性质的先验知识，使用一种成为逐层搜索的迭代方法，其中 k 项用于探索（k+1）项集。其核心是基于两阶段频集思想的递推算法。该关联规则在分类上属于单维、单层、布尔关联规则。在这里，所有支持度大于最小支持度的项集称为频繁项集，简称频集。

Apriori 算法的基本思想是：首先找出所有的频集，这些项集出现的频繁性至少和预定义的最小支持度一样。然后由频集产生强关联规则，这些规则必须满足最小支持度和最小可信度。然后使用上步找到的频集产生期望的规则，产生只包含集合的项的所有规则，其中每一条规则的右部只有一项，这里采用的是中规则的定义。一旦这些规则被生成，那么只有那些大于用户给定的最小可信度的规则才被留下来。为了生成所有频集，使用了递归的方法。

$Aprior(T,\epsilon)$

$L_1 \leftarrow \{large\ 1-itemsets\}$

$k \leftarrow 2$

$while L_{k-1} \neq \emptyset$

$C_k \leftarrow \{a \cup \{b\} | a \in L_{k-1} \wedge b \notin a\} - \{s | s \subseteq c \wedge |s| = k-1\} \notin \{L_{k-1}\}$

$for\ transactions\ t \in T$

$C_t \leftarrow \{c | c \in C_k \wedge c \subseteq t\}$

$for\ candidates\ c \in C_t$

$cout[c] \leftarrow cout[c] + 1$

$L_k \leftarrow \{c | c \in C_k \wedge cout[c] \geq \epsilon\}$

$k \leftarrow k + 1$

$return\ \cup_k L_k$

（2）后缀数组

在字符串处理当中，后缀树和后缀数组都是非常有力的工具，其中后缀树大家了解得比较多，关于后缀数组则很少见于国内的资料。其实后缀数组是后缀树的一个非常精巧的替代品，它比后缀树容易编程实现，能够实现后缀树的很多功能而时间复杂度也不太逊色，并且，它比后缀树所占用的空间小很多。可以说，后缀数组是一种解决字符串问题的有力工具。相比于后缀树，它更易于实现且占用内存更少。在实际应用中，后缀数组经常用于解决字符串有关的复杂问题。

2.3 本文的贡献

基于上述研究基础和算法基础，结合具体的 DNS 日志情况，本文改进并实现了基于 Apriori 的 DNS 访问流量和路径频繁性分析技术。其具体贡献如下：本文改进并实现了基于 Apriori 的频繁连续时间片段选择算法，针对所有频繁访问的域名，挖掘其流量随时间的变化情况，继而可以作出流量分析、异常检测、攻击预测等。 本文采用基于 Apriori 的频繁域名解析路径分析算法，对现有的域名解析路径数据分析，挖掘出域名解析路径中的频繁项集，以此作为约束条件来优

化服务器部署，缓解服务器拥堵等情况。

第三章 基于 Apriori 的频繁连续时间片段选择算法

3.1 算法概述

如第一章所述，用户对于不同域名的访问具有一定的时间规律特征。为了验证这一结论，我们对连续七天的 DNS 访问信息进行了系统的统计，发现其七天的单位时间流量变化趋势基本一致。基于这个结论，我们以网上所提供的连续 7 天的 DNS 访问日志为数据集，在日志基础上首先进行数据清理，清除时间戳错误的数据信息；然后对 7 天连续频繁访问的数据进行单位时间的流量统计，分析并以特定的字符形式纪录流量随时间变化趋势；对已纪录的字符串信息，采用 Apriori 和后缀数组匹配相结合的方式进行分析，筛选出具有连续时间的时间变化序列的频繁序列。

这些频繁序列反映了用户对该域名的访问情况随时间变化的通性规律，既每天某几个连续时间内，对该域名的访问都会有该规律出现。因此，我们可以将分析结果用于异常检测、攻击预测等——当某一天对该域名的访问趋势与该规律出入较大时，我们可以认为此时发生了网络异常。特别的，当出现访问流量剧变时，认为出现了网络攻击（如 DOS 攻击）。

3.2 实现步骤

本算法的实现步骤及具体细节如下，每一步的操作结果详见附属文件夹：

- 数据清理。

对每一天的会话纪录进行统计，对日志分析可以发现，其时间戳应该是顺序的，但是在整点转换时发现，会话纪录会出现乱序现象——例如前一个时刻为 14:59:59，第二个时刻为 15:00:01，而第三个时刻为 14:59:58，第四条纪录为 15:00:02。这种整点转换处的时间戳混乱现象，会对单位时间的流量统计结果产生较大的误差，为了降低统计误差，我们采取的措施为：如果某两条小时数相等的会话纪录

中间夹杂了其他小时数的纪录,则认为这些夹杂的纪录为错误数据,将其删除(如本例中，删除第三个时刻的会话纪录)。

• 找出每天频繁访问的域名。

对每天的会话纪录进行分析，统计每个域名在当天会话纪录中的访问次数，并设置阈值 M，当该域名的访问次数大于 M 时，认为该域名的访问是频繁的，纪录其域名和对应的访问次数。针对本数据源，我们的综合分析，将阈值 M 设置为 250（因为绝大多数的频繁项访问次数在 250-600 之间)，统计结果详见文件夹“Session data result（每天访问量大于 250 的域名)”。

• 七天的频繁域名信息集成。

对连续七天的“频繁域名”进行求交集运算，得到连续七天中，一直被频繁访问的域名。对本数据源数据进行分析，共得到了 23 个频繁访问的域名（结果见文件“Intersect_Url（7 天内所有访问量高的域名)”)。

• 统计各频繁域名的单位时间访问流量。

对各个域名分别统计每天单位时间的访问流量。结合具体数据集情况和分析需求，我们将时间单位设置为单位小时，分别统计每个小时的改域名的访问次数作为其流量，23 个频繁域名，每个域名可以得到 1 个文件，文件内记录该域名七天内，每天 24h 单位小时的访问请求流量，结果详见文件夹“result_perhour（各域名的每小时访问量)”。 • 对各域名标记流量变化序列。对 23 个频繁域名的单位小时流量进行分析，并对其访问量的变化情况用 u、s、d 三个字符和当前时间戳的小时数予以标记。具体标记策略为：将某一小时的访问量与前一小时的访问量做比较，得到差值 d。设置阈值 D>0,如果 d>D，则记录字符串“T(h)”(时间戳的小时数部分)＋“u”(up)；如果 d<-D，则记录字符串“T(h)”(时间戳的小时数部分)＋“d”(down)；如果｜d｜<=D；则记录字符串“T(h)”(时间戳的小时数部分)＋“s”(steady)。特别地，对于每天的第一个小时，我们记录为 s。

经过反复测试，我们发现，当阈值 D>15 时，各个时间段的流量状态基本都为 s，这就失去了数据挖掘的意义；而如果阈值设置过小，很小的波动和很大的访问量变化都会被予以同样的“u”或“d”标记，这也会同样导致挖掘的数据不

能很好的反映整体的变化趋势，因此作为折中，我们将阈值 D 设置为 10。通过此方法对 23 个频繁域名的访问情况标记记录，得到 23 个文件，结果如文件夹“result D（各域名的 s u d 序列)”所示。

- 挖掘各频繁域名的频繁项集。

分别对各个域名七天的 usd 序列进行分析，采用 Apriori 算法和后缀数组名相结合的方式进行频繁序列的挖掘，得到各个域名的 usd 频繁序列。

与已有的方法不同，我们并不直接采用 Apriori 算法进行频繁项挖掘，相反，首先，对所有的频繁域名用后缀数组的方法，挖掘出 7 天共有的 usd 最长序列项，得到最长的频繁序列长度 L。然后对该域名的以 Apriori 算法，挖掘出长度小于 L 的所有频繁序列。两种方式相结合，以剪枝的策略进行搜索，避免 Apriori 算法对于过长频繁序列的计算判断，可以大大提高算法效率。

- 频繁序列筛选。

因为上述步骤所得到的频繁序列无序，且不能保证时间戳的连续性。为此，本步骤需要在其结果中进行筛选，留下时间戳连续且间隔为 1 小时递增的 usd 频繁序列。执行结果详见文件夹“Final_result(各域名频繁项集)”。

通过以上几个步骤，最终得到 DNS 日志记录中的“频繁域名”访问的 usd 频繁序列，这些序列可以反映出该域名在对应时间段内的请求访问量增减趋势，反映了用户对该域名的访问情况随时间变化的通性规律，既每天某几个连续时间内，对该域名的访问都会有该规律出现。因此，我们可以将分析结果用于异常检测、攻击预测等——当某一天对该域名的访问趋势与该规律出入较大时，我们可以认为此时发生了网络异常。特别的，当出现访问流量剧变时，认为出现了网络攻击（如 DOS 攻击)。

3.3 算法要点

与已有的方法不同，我们并不直接采用 Apriori 算法进行频繁项挖掘，相反，首先，对所有的频繁域名用后缀数组的方法，挖掘出 7 天共有的 usd 最长序列项，

得到最长的频繁序列长度 L。然后对该域名的以 Apriori 算法，挖掘出长度小于 L 的所有频繁序列。两种方式相结合，以剪枝的策略进行搜索，避免 Apriori 算法对于过长频繁序列的计算判断，可以大大提高算法效率。

第四章 基于 Apriori 的频繁域名解析路径分析算法

本章节主要介绍了对域名解析路径进行的数据分析，通过对域名解析路径数据分析，挖掘出域名解析路径中的频繁模式，用于优化服务器部署，避免出现某一服务器过“过忙”的现象。

4.1 算法概述

为了得到域名路径解析的频繁项集，我们首先对数据进行清理，得到较为干净可靠的筛选数据；对于频繁项集的获取，我们采用基于 Apriori 的频繁域名解析路径分析算法，对现有的域名解析路径数据分析，挖掘出域名解析路径中的频繁项集。我们对每天的会话访问数据集进行分析，挖掘出该天的服务器域名解析的频繁项集。当某一解析路径频繁出现，同时该路径上某一服务器的访问流量过大时，就可以重新部署该路径上的服务器，将改路径更改至经过某一较空闲的服务器上，从而缓解服务器压力，优化服务器部署。

4.2 实现步骤

本算法的实现步骤及具体细节如下：

- 数据清理。

使用 3.2 节中使用的数据清理规则对每天的会话记录中的错误数据进行清除。

- 数据处理。

在会话访问数据集中，统计记录第 t 天记录中出现的所有 IP 地址(假设共有个 N 个 IP 地址)，对于每条访问记录，我们采用 0/1 标记的方式，对于出现的 IP 地址标记为 1，因此，我们将每天的会话记录转化为的 0/1 矩阵形式，方便后续对数据的处理。

- 频繁项集的获取。

分别对 7 天的处理后的路径解析数据进行处理，采用 Apriori 算法对数据集进行处理得到域名服务器域名解析路径中的频繁项集，经过反复试验测试，最终我们取参数 sup=0.5% ,con=1%来获取最终数据。

通过上述步骤，我们可以从域名解析日志中挖掘出域名解析路径频繁模式。这些序列可以表示出在某天内频繁解析的域名解析路径。通过从域名解析日志中挖掘出频繁模式，可以对现有的服务器部署方案进行改进，优化服务器部署，防止网络“拥堵”等异常情况的产生。

第五章 结果分析

5.1 基于 Apriori 的频繁连续时间片段选择算法实现

5.1.1 数据源及实现环境

本技术实现的数据源和实现环境如表 1 所示

数据源	DNS 服务器连续 7 天访问日志
实现系统	windows 7
实现语言	java（Apriori），c++（后缀数组）
jdk 版本	jdk1.7
分析工具	clementine

表 1 数据源和实现环境表

5.1.2 频繁序列挖掘结果举例

根据第三章的基于 Apriori 的频繁连续时间片段选择算法，在 5.1.1 的数据集合环境基础上，对数就进行处理、挖掘和分析，并得到频繁域名的频繁 usd 序列如附属文件夹。为方便读者理解，更清晰的表现出实现结果，现以域名 c.wanfangdata.com.cn 为例，解释实现过程和结果：

首先进行清理，然后找出每天访问次数大于 250 次的域名，并对七天的频繁域名信息集成进行，得到文件 Intersect_Url（7 天内所有访问量高的域名）。 然后统计各频繁域名的单位时间访问流量。以域名 c.wanfangdata.com.cn 为例，得到其七天每天每个小时的访问次数，如下表 2 所示：

天\小时	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	86	87	76	88	75	13	87	82	83	78	84	78	83	90	79	85	76	84	96	91	77	90	89	86
2	49	10	15	3	19	16	18	16	11	90	93	87	85	91	87	98	90	91	96	88	80	74	83	72
3	92	88	78	78	84	83	74	81	87	89	94	97	94	91	88	93	92	88	97	84	85	89	78	84
4	104	101	108	100	104	99	104	103	102	100	101	103	108	105	99	100	104	105	102	102	100	103	102	103
5	80	86	77	79	76	78	82	86	83	89	94	90	85	86	93	91	92	87	93	93	95	82	84	83
6	76	83	79	76	83	82	76	81	88	85	92	83	91	94	93	87	84	93	94	77	85	91	79	79
7	83	77	80	78	89	16	84	82	77	72	84	83	87	78	71	81	78	91	95	80	79	89	83	85

表 2 c.wanfangdata.com.cn 七天单位小时的访问次数

对表 2 中的数据进行处理，并对其访问量的变化情况用 u、s、d 三个字符和当前时间戳的小时数予以标记。以域名 c.wanfangdata.com.cn 为例，可以得到如表 3 所示七天流量变化 usd 序列：

天数	usd 序列
1	0s 1s 2d 3u 4d 5d 6u 7s 8s 9s 10s 11s 12s 13s 14d 15s 16s 17s 18u 19s 20d 21u 22s 23s
2	0s 1d 2s 3d 4u 5s 6s 7s 8s 9u 10s 11s 12s 13s 14s 15u 16s 17s 18s 19s 20s 21s 22s 23d
3	0s 1s 2s 3s 4s 5s 6s 7s 8s 9s 10s 11s 12s 13s 14s 15s 16s 17s 18s 19d 20s 21s 22d 23s
4	0s 1s 2s 3s 4s 5s 6s 7s 8s 9s 10s 11s 12s 13s 14s 15s 16s 17s 18s 19s 20s 21s 22s 23s
5	0s 1s 2s 3s 4s 5s 6s 7s 8s 9s 10s 11s 12s 13s 14s 15s 16s 17s 18s 19s 20s 21d 22s 23s
6	0s 1s 2s 3s 4s 5s 6s 7s 8s 9s 10s 11s 12s 13s 14s 15s 16s 17s 18s 19d 20s 21s 22d 23s
7	0s 1s 2s 3s 4u 5d 6u 7s 8s 9s 10u 11s 12s 13s 14s 15s 16s 17u 18s 19d 20s 21s 22s 23s

表 3 c.wanfangdata.com.cn 七天流量变化 usd 序列

然后依据表 3 中数据，分别对各个其七天的 usd 序列进行分析，采用 Apriori 算法和后缀数组名相结合的方式进行频繁序列的挖掘，得到该域名的 usd 频繁序列，并要在其结果中进行筛选，留下时间戳连续且间隔为 1 小时递增的 usd 频繁序列，如表 4 所示：

二元频繁序列	11s 12s
	12s 13s
	7s 8s
三元频繁序列	11s 12s 13s

表 4 c.wanfangdata.com.cn 的频繁序列

23 个频繁访问的域名，最终共得到 23 个类似于表 4 的频繁序列表。根据这些表格中的频繁序列可以反映出该域名在对应时间段内的请求访问量增减趋势，反映了用户对该域名的访问情况随时间变化的通性规律，即每天某几个连续时间内，对该域名的访问都会有该规律出现。因此，我们可以将分析结果用于异常检测、攻击预测等——当某一天对该域名的访问趋势与该规律出入较大时，我们可以认为此时发生了网络异常。特别的，当出现访问流量剧变时，认为出现了网络攻击（如 DOS 攻击）。

5.2 基于 Apriori 的频繁域名解析路径分析算法

表 5 表示第四章中讲述算法从域名解析日志中挖掘出域名路径解析频繁模式，按照支持度排序(取前 10 位)。

Consequent	Antecedent	Support %	Confidence %
202.108.44.55 = 1	202.106.184.166 = 1	4.041	13.201
61.172.201.254 = 1	202.106.184.166 = 1	4.041	6.243
121.14.1.22 = 1	202.106.184.166 = 1	4.041	4.455
202.106.184.166 = 1	202.108.44.55 = 1	3.947	13.514
61.172.201.254 = 1	202.108.44.55 = 1	3.947	7.432
121.14.1.22 = 1	202.108.44.55 = 1	3.947	4.505
168.160.96.25 = 1	168.160.184.101 = 1	3.65	2.619
216.239.36.10 = 1	216.239.34.10 = 1	3.358	4.004
216.239.32.10 = 1	216.239.34.10 = 1	3.358	3.971
216.239.38.10 = 1	216.239.34.10 = 1	3.358	2.449

表 5 域名路径频繁模式（按照支持度排序）

表 6 表示第四章中讲述算法从域名解析日志中挖掘出域名路径解析频繁模式，按照置信度排序(取前 10 位)。

Consequent	Antecedent	Support %	Confidence %
121.196.255.97 = 1	121.196.255.98 = 1 and 121.196.255.148 = 1	0.545	13.201
121.196.255.148 = 1	121.196.255.98 = 1 and 121.196.255.97 = 1	0.545	6.243
121.196.255.97 = 1	121.196.255.148 = 1 and 112.126.125.147 = 1	0.545	100.0
121.196.255.148 = 1	121.196.255.98 = 1 and 121.196.255.147 = 1	0.543	100.0
121.196.255.98 = 1	121.196.255.148 = 1 and 121.196.255.147 = 1	0.543	100.0
121.196.255.148 = 1	121.196.255.98 = 1 and 112.126.125.147 = 1	0.543	100.0
121.196.255.148 = 1	121.196.255.98 = 1 and 112.126.125.148 = 1	0.543	100.0
121.196.255.98 = 1	121.196.255.148 = 1 and 112.126.125.148 = 1	0.543	100.0
121.196.255.97 = 1	121.196.255.98 = 1 and 121.196.255.147 = 1	0.543	100.0
121.196.255.97 = 1	121.196.255.98 = 1 and 112.126.125.147 = 1	0.543	100.0

表 6 域名路径频繁模式（按照置信度排序）

从域名解析日志中挖掘出域名解析路径频繁模式。这些序列可以表示出在某天内频繁解析的域名解析路径。通过从域名解析日志中挖掘出频繁模式，可以对现有的服务器部署方案进行改进，优化服务器部署。

第六章 展望

本文在第三章介绍的基于 Apriori 的频繁连续时间片段选择算法将 Apriori 和后缀数组匹配相结合的方法，从域名解析的路径数据中挖掘出用户访问 DNS 的流量信息，筛选出数据中每个时间段之间的频繁项集，根据连续时间段的域名服务器流量变化来推测下一时间段的服务器流量。通过预测得到的服务器流量与真实流量数据做比较，以此来判断服务器状态，推测出服务器正常与否。鉴于数据规模的局限性，在给定较少的数据集的情况下不能得到较多的流量变化规律；假定给定的数据集足够大，数据信息足够充分，该方法可以较好的挖掘出服务器的流量变化规律；该方法单纯从服务器流量变化的角度出发，在不使用其他监控软件的情况下，通过得到的频繁项集序列得到的域名访问情况变化规律来预测服务器的工作状态。

本文在第四章中介绍的基于 Apriori 的频繁域名解析路径分析算法，通过 Apriori 算法从域名解析日志中挖掘出域名解析路径频繁模式。鉴于数据规模的局限性，在给定较少的数据集的情况下不能得到较多的域名解析路径。假设在给定的数据集足够大，数据信息足够充分，该方法可以较好的挖掘出服务器域名解析路径频繁模式。从而可以得到更好的域名解析路径频繁项集来对服务器部署进行优化。

第七章 总结

此工作我们小组三个人一起查阅资料，经过详细讨论、认真分析，集思广益，在已有的研究基础之上提出了本文的创新所在。大家一起努力，团结合作，共同完成了本技术的实现和报告。具体任务分工安排如下： 苏思悦 郑越：查阅文献，阅读论文，主要负责改进并实现了基于 Apriori 的频繁连续时间片段选择算法模块，包括数据处理、创新点挖掘、算法分析实现、论文撰写，同时对源代码和结果数据进行整理。 王恒怿：查阅文献，阅读论文，主要负责基于 Apriori 的频繁域名解析路径分析算法模块，包括数据处理、创新点挖掘、算法分析实现、论文撰写，并整合技术文档。

参考文献

[1] Cheng J, Li X, Yuan J et al.(2010) K-means based analysis of DNS query patterns.J Tsinghua Univ 17:80 − 87.

[2] Wang Z, Li X, Yan B(2010). Abnormity detection of DNS query traffic at CN top level domain server, Technical Report. http://www.docin.com/p-629288242.html.

[3] Hongyuan Cui, Jiajun Yang, Ying Liu, Zheng Zheng, Kaichao Wu. Data Mining-based DNS Log Analysis, Sci.(2014)1(3-4):311-323.