使用机器学习的自动流量分类和应用程序识别

Sebastian Zander, Thuy Nguyen, Grenville Armitage

斯威本科技大学 高级互联网架构中心，澳大利亚 墨尔本

摘要

负责网络流量跟踪的动态分类器和应用程序识别器对 IP 网络设计、管理和监控中的一些关键领域提供了很多好处。目前，这样的分类器依靠选定的数据包报头字段（例如端口号）或应用程序层协议来解码。这些方法有许多不足，例如，许多应用程序使用不可预知的端口号和协议解码，这需要大量的计算资源，或者，一旦协议是未知的或加密的，这就是不可行的。本文使用无监督的机器学习技术，提出了一种新方法用于流量分类和应用程序识别。基于统计流动特性，流量可以自动分类。我们使用收集来的不同网络区域的几条流量轨迹中的数据，评估提出方法的效率。我们使用特征选择找到一个最佳的特性集，并确定不同特征的影响。

1.简介

近年来，使用互联网的应用程序的种类已经大幅增加。除了“传统的”应用程序（如电子邮件、网页或 ftp），新的应用程序也有了强劲的势头（如流媒体、游戏或 P2P）。根据应用程序，动态识别和分类流量的能力十分有利于:

- 趋势分析（估计网络设计的需求趋势大小和起源的能力）
- 适应性，基于网络的流量标志需要特定的没有直接客户端应用程序或终端主机参与的服务质量（QoS）
- 动态访问控制（可以检测禁止应用程序、拒绝服务攻击(DoS)或其他不必要的流量的自适应防火墙）
- 合法的窃听（实现最低入侵保证和基于流量细节统计总结的窃听）

■ 入侵检测（检测与恶意用户或蠕虫导致的安全漏洞相关的可疑活动）

一些常见的基于检查已知端口号的识别技术不再精准，因为许多应用程序不再使用固定的、可预测的端口号。互联网地址分配机构（IANA）[1]分配从0－1023的著名端口号，注册从1024-49151不等的端口号。但许多应用程序没有IANA分配或注册的端口号，只使用“众所周知”的缺省端口。通常，缺省端口号和IANA端口号重叠，使用这些端口号时，在网络中不会有明确的身份[2]。甚至，缺省端口或注册了端口号的应用程序最终也会使用不同的端口号，因为（1）非特权用户经常需要使用1023以上的端口号，（2）用户可能会故意试图隐藏他们自身的存在，或绕过出口过滤器，（3）多个服务器共享一个唯一的IP地址（主机）。此外，某些应用程序（如，被动FTP或视频/语音通讯）使用提前不可知的动态端口。

在许多当前的工厂生产中，使用的一个可靠技术涉及来自数据包的会话和应用程序信息的有状态的重建（例如[3]）。尽管这种技术避免了依赖固定的端口号，但是它十分复杂，强加了处理加载流量识别的设备。它必须用大量的应用程序语义和网络级语法的知识保持更新，必须足够强大，能够执行潜在的大量流量的并发分析。当在处理专有协议或加密流量时，这种方法是困难和不可行的。另一个问题是，直接分析会话和应用程序层的内容，可能代表明确违反组织隐私政策，或违反有关隐私的立法。[4]的作者提出，基于签名的方法对P2P流量进行分类。虽然这些方法比状态重建更有效率，比基于端口的方法提供更好的分类，但是它们仍然依赖协议。[5]的作者描述了一个绕过基于协议检测器的方法。

以前的工作中，使用许多不同参数描述网络流量（例如，[4]、[6]、[7]），包括流量的大小、持续时间、包长度、间隔时间的分布、流量闲置时间等。我们提出使用机器学习（ML）[8]，在这些参数的基础上，自动分类和识别网络应用程序。通过学习取决于特征的数据集的固有结构，机器学习算法可以自动生成一个分类器。在过去的十年中，机器学习已经从在实验室演示转变为具有显著的商业价值[9]。我们的方法包括识别流量属性最优集合的任务，这能最小化

处理代价，最大化分类精准度。使用收集来的不同网络区域中的流量轨迹，来评估我们方法的效率。

本文的其余部分安排如下。第 2 节是相关工作的概述。第 3 节介绍提出的基于机器学习的应用程序识别方法。第 4 节使用流量轨迹，评估方法的效率。第 5 节总结和概述下一步工作。

2.相关工作

使用机器学习技术对流量分类的想法，在入侵检测[10]的文章中第一次被引入。

[11]的作者用主成分分析（PCA）和密度估计把流量分类到不同的应用程序。他们使用来自一个较小数据集的两个流量属性的分布，研究了一些众所周知的端口。

在[12]中，作者使用近邻取样（NN）和线性判别分析（LDA）方法，用不超过四个属性，成功地把应用程序映射到不同的 QoS 等级中。使用这种方法，类别数目小，且已知先验。汇总用于这种分类的属性花了超过 24 小时。

[13]使用了期望最大化（EM）算法，这种算法使用固定属性集，把流量分类到不同的应用程序类别中。作者发现，该算法分离流量到几个基本类别，但是从他们的评估来看，目前尚不清楚不同的属性和 EM 参数会有什么影响、这种分类实际应用中会有多好。

在[14]中，作者使用模拟退火的 EM 算法，根据流量大小（如老鼠和大象），对流量分类。作者认为，他们的方法比以前基于阈值的方法，产生了更有意义的结果。

在[15]中，我们已经提出了一个基于机器学习的识别不同网络应用的方法。在本文中，我们使用从互联网中不同区域收集到的许多流量轨迹，来评估这个方法。

[16]的作者用了一个基于朴素贝叶斯分类器的类似方法，也用了大量的流量属性。他们只用一个数据集，但数据集中的流量已经被手工分类，这就会得到一个非常准确的评估。

文献[17]提出，结合不同的非机器学习技术来识别网络应用程序。

3. 基于机器学习的应用识别

首先，我们把数据包分成双向流量，使用 NetMate（网络流量采集及分析软件）[18]计算流量的特征。选择流量数据的一个子集用作样本，来提高学习过程的性能（见[15]的详细介绍）。然后，用流特征和一个流属性的模型学习类别（1）。一旦已经学习了所有类别，就可以分类新流量（2）。

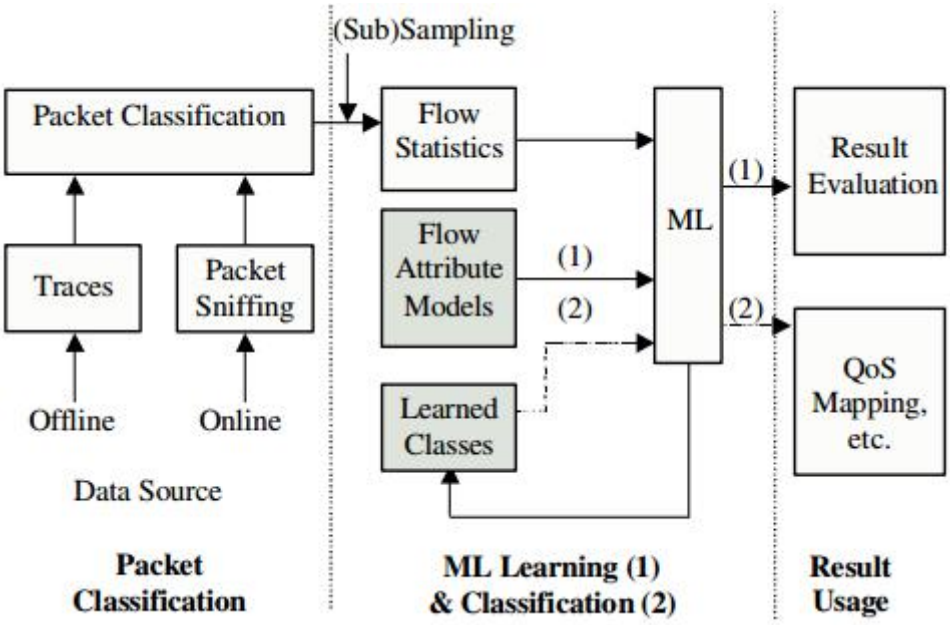

图 1：基于机器学习的流量分类

注意，我们使用术语学习用于创建分类器的首次过程，用来区分它和上一个分类器的过程。然而，在其他工作中，这有时也称为分类或聚簇。可以导出学习和分类的结果用于评估。分类的结果能用于例如 QoS（服务质量）映射、趋势分析等。

3.1 机器学习算法

对于机器学习，我们使用自动聚类算法[19]。自动聚类是一种无监督贝叶斯分类器，它能用训练数据集中未分类的样本，基于样本的一些属性，自动学习数据集固有的“自然”类别（也称为聚类）。然后，可以用得到的分类器对未见过的新样本进行分类。在本节中，我们只提供一个概述，建议读者阅读[19]中算法的详细描述。

在自动分类模型中，所有样本实例都必须条件独立，因此，任何两个相似的实例由其中一个占据该类别。一个实例所属的类别是每个实例的未知或隐藏属性。

集合有 I 个实例（数据空间），一个实例 X_i 的属性记为$\vec{X}i$，类别集合中有 J 个类，X_i 属于一个特定的 C_j 类的概率由两部分组成：类间概率和每个类的概率密度函数（类内概率）。因为不同类别构成了数据的离散分区，合适的类间概率密度函数是伯努利分布，它以概率为$\{\pi_1, \cdots, \pi_j\}$的集合$\vec{V}j$作为特征，限制 $0 \leqslant \pi_j \leqslant 1$，$\sum_j \pi j = 1$。因此

$$P(X_i \in C_j \mid \vec{V}_c) \equiv \pi_j \qquad (1)$$

类间概率只依赖于 J，不在$\vec{X}i$上的已知数量的实例被分配给 C_j。类内概率是 k 个属性的条件独立概率分布的乘积：

$$P(\vec{X}_i \mid X_i \in C_j, \vec{V}_j) = \prod_k P(X_{ik} \mid X_i \in C_j, \vec{V}_{jk}) \quad (2)$$

自动分类对单个 $P(X_{ik})$ 支持离散的和实值的属性模型。然而，我们只使用真正的属性，这些属性用对数正态分布建模。因此，我们假设，属性的概率密度函数只有一个函数形式，并在所有方程中省略该参数（更一般的方法见[19]）。结合类间和类内概率，我们得到了属性值为$\vec{X}i$的实例 X_i 属于 C_j 类的概率为：

$$P(\vec{X}_i, X_i \in C_j \mid \vec{V}_c, \vec{V}_j) = \pi_j \prod_k P(X_{ik} \mid X_i \in C_j, \vec{V}_{jk}) \quad (3)$$

通过在参数集上引入先验，可以被转换成一个贝叶斯模型，并获得参数集$\vec{V}$和当前数据空间 X 的联合概率：

$$P(X\vec{V}) = P(\vec{V})P(X \mid \vec{V}) \quad (4)$$

目标是从参数后验概率密度函数找到最大后验参数值：

$$P(\vec{V} \mid X) = \frac{P(X, \vec{V})}{P(X)} = \frac{P(X, \vec{V})}{\int d\vec{V} P(X, \vec{V})} \quad (5)$$

将数据划分成 J 个非空的子集，直接使用这个方程，计算每一部分的后验概率。但 J 较小时，可能的划分数接近 J^I，这使该方法计算大量实例和/或类的集合是不可行的。因此，自动分类用基于 EM 算法[20]来近似。自动分类在估算所有实例类别任务和估计参数$\vec{V}$之间循环。EM 算法保证收敛到局部最大值。在试图找到局部最大值时，自动分类算法从参数空间中的伪随机点开始，执行重复 EM 搜索。在给出当前数据空间的条件下，参数集有最高概率的模型被认为是最好的。

可以用类的数量（如果已知）预先配置自动分类算法，或者算法可以试着估计类的数目。对于我们的问题来说，确切的类的数目是事先未知的。有人可能认为，每个应用程序应该只有一个类。然而，我们发现，流量属性的分布——即使是一个应用程序——也是相当复杂的。由于我们使用的是简单的属性模型（对数正态分布），用一个类建模一个应用程序是不可能的。当类别数目未知时，用类别数目为$\vec{J}_{start}$的开始列表配置自动分类算法。然后，对于每次 EM 搜索，只要有剩余输入，类的初始数目取自下一个输入的$\vec{J}_{start}$。如果一些类不收敛，在一次 EM 搜索结束后，类的数目会更小。对于开始列表之后的所有迭代，会耗尽自动分类算法随机选择的来自对数正态分布的 J，目前发现最适合的类数目是 10 种。

每个应用有多个类别，这给应用程序提供了一个更细粒度的视图的好处。例如，网页流量用于不同的目的（如，批量传输、交互式接口、流等），进行详细分析，区分它们也是十分有益的。另一方面，在运行时间和内存上，类的数目越多就会降低方法的性能。

3.2 特征选择

我们的特征选择技术基于学习算法的实际性能。 一般情况下，这个方法能达到最高精确度，因为它为算法“裁剪”特征集。缺点是比算法无关的方法，例如基于相关的特征选择（CFS）[21]，计算花费大（特别是当学习算法不是非常快的时候）。

找到提供应用程序类别反差最大的属性组合，是一个重复过程：（1）选择一个属性子集，（2）学习类别，（3）评估类的结构。我们实现了用序列前向选择（SFS）找最佳的属性集合，因为穷举搜索是不可行的。该算法开始于每一个单一属性。产生最好结果的属性，放于选定的属性列表 SEL（1）中。然后，结合 SEL（1）中的所有属性，尝试不在 SEL（1）中的第二个属性。能产生最好结果的属性组合成为 SEL（2）。重复该过程，直到没有进一步的提高。SFS 只是一个（简单的）方法来确定最有用的特征集，还有其它的方法，如，序列后退消除（见[22]）。

为了评估分类结果的质量，我们提出一种称为类内同质性 H 的度量标准。我们把各个学习期间的应用程序和类别集合定义为 A 和 C。我们还定义了函数 count(a,c)，该函数计算流量的数量，其中，应用程序 a∈A，所属类 c∈C。然后，一个类 c 的同质性 H(c)被定义为类中一个应用的流量的最大部分：

$$H(c)=\frac{\max(count(a,c)\,|\,a\in A)}{\sum_{a} count(a,c)} \qquad (6)$$

类的总体同质性 H 是各个类的同质性的均值：

$$H = \frac{\sum_c H(c)}{\|C\|} \quad \text{and} \quad (0 \le H \le 1) \qquad (7)$$

目标是最大化 H，从而在不同的应用程序之间有好的划分。 我们不用标准指标，比如准确度、精确度和查全率，使用 H 作为评估指标的原因是，我们使用的是无监督学习技术。类的数目和类到应用程序的映射在学习前是未知的。之后，在某类中有最多流量的应用程序分配到该类。然而，如果不止一个应用贡献了类中的大量流量，那么得到的映射是不好的。为了明确地将一个类映射到一个应用，需要高同质性的值。

4. 评估

4.1 跟踪文件

对于评估，我们使用来自 NLANR[23]的 Auckland-VI、 NZIXII 和 Leipzig-II 的轨迹，它们在不同年代，在因特网的不同位置被捕获。因为我们受限于使用公开获得的匿名轨迹，所以我们无法验证创建流量的真实应用程序。因此，在我们的评估中，假设流量被定义的 IANA 或注册的服务器端口识别应用程序。在我们的例子中，服务器端口通常是双向流动的目的端口。有极少数情况，源端口是 IANA 定义的端口，我们已经交换了双方流动的方向（包括 IP 地址，端口和流量属性）。

我们承认，假设服务器端口始终能识别出应用程序是不正确的。然而，我们假设，对于我们在本研究中使用的端口来说，多数的流量来自预期应用。然后，小“错误”的流量最有可能降低学习到的类的同质性。因此，我们的评估结果可以视为有效性的下界。除了标准服务器端口，我们也不考虑其他端口中的选择的应用的流量，例如，我们只考虑端口 80 中的网页流量，而不是端口 81 中的。假设使用的服务器端口和应用属性之间没有很强的相关性，这不会引入任何附加的偏差，因为可以看作随机抽样。

4.2 流量属性

属性设置包括数据包到达的间隔时间、数据包长度均值和方差、流量大小（字节）和持续时间。除了持续时间，所有属性都是双向的，意味着它们要在流量的两个方向上被计算。我们的目标是尽量减少属性的数量，只用容易计算的“基本”属性。我们没把服务器端口作为一个属性，因为“错误”的端口会引入一个未知偏差。

在我们的分析中，我们排除了每个方向少于三个数据包的流量，因为对很短的流量，只能计算某些属性，例如，仅有一个数据包的流量，没有到达间隔时间和数据包长度的统计数据，只能计算前向方向。超过一半的可用属性数量将难以分离不同的应用，最可能使属性影响结果的偏差。此外，任何有效的 TCP 流量应该有至少六个数据包。然而，有效的 UDP 流量可以只包含两个数据包，因此，除了小 UDP 流，其他的可能对 DNS 和半生期流量产生结果偏差。

这种策略显然留下了许多流量，但在此工作中，我们的目标是分离不同的应用程序，我们不对“奇怪的”流量或异常的流量感兴趣。实际上，利用这些流量很危险，因为验证我们的方法需要依赖服务器端口。例如，如果使用含一个数据包的流量，我们可能会混淆端口扫描和真实的应用程序。然而，通常来说，小流量能提供让人感兴趣的结论，特别是从安全角度考虑，小流量不应该被忽略。

4.3 识别网络应用

出于性能的考虑，我们使用每个跟踪文件的一个子集，包含 8000 个流量。对于每个应用程序（FTP 数据、远程登录、SMTP、DNS、HTTP、AOL 即时通讯、Napster、半减期）来说，我们从特定应用程序的所有流量中随机抽取 1000 个流量。我们获得流量样本，对于四条轨迹中的每一条，使用学习算法的同一参数执行 SFS。

图 2 显示了用固定属性集运行一次算法的样例的结果。它显示了所有应用程序在已知的类别中是如何分布的。所有类从左向右类大小减少（类别数增加）

排序。对于每个类，同质性是一个应用程序流量的最大部分，例如，H(最左边)= 0.52，H(最右边)= 1。整体同质 H 是所有类同质性的均值，在此例中，H =0.86。

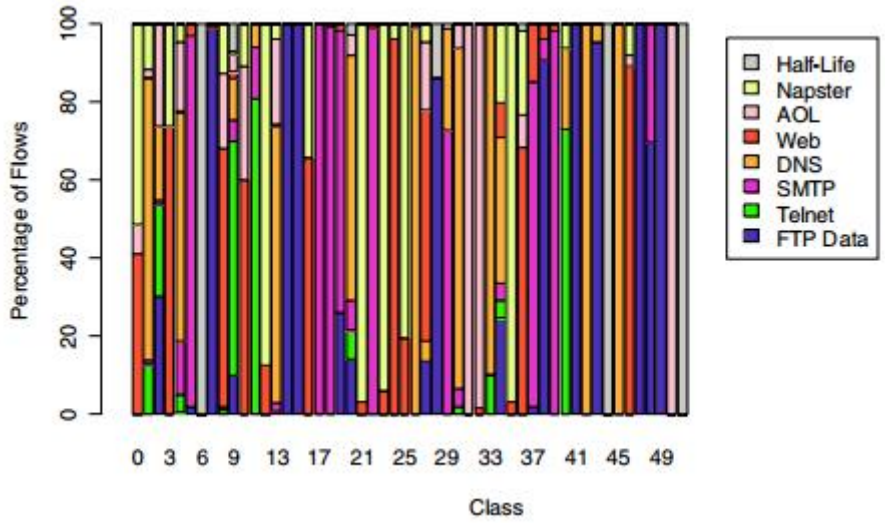

图 2：跨类别应用程序的样例分布

图 3 展示了总体均值 H 和属性数目的依赖关系。它显示了识别不同应用的整体效率随着属性数目的增加而增加，直到它达到最大值，该值取决于轨迹，在 0.85 和 0.89 之间。这意味着，平均来看，分离不同应用程序的流量有相当高的效率。最终的最佳集合中属性的数量在四和六之间变化，但总是比在 SFS（十一个）中使用的属性总数量小很多。我们也在整个特征集上对所有轨迹进行了训练，但没有发现同质性得到显著改善（值在 0.85 和 0.90 之间），而且学习过程十分缓慢。

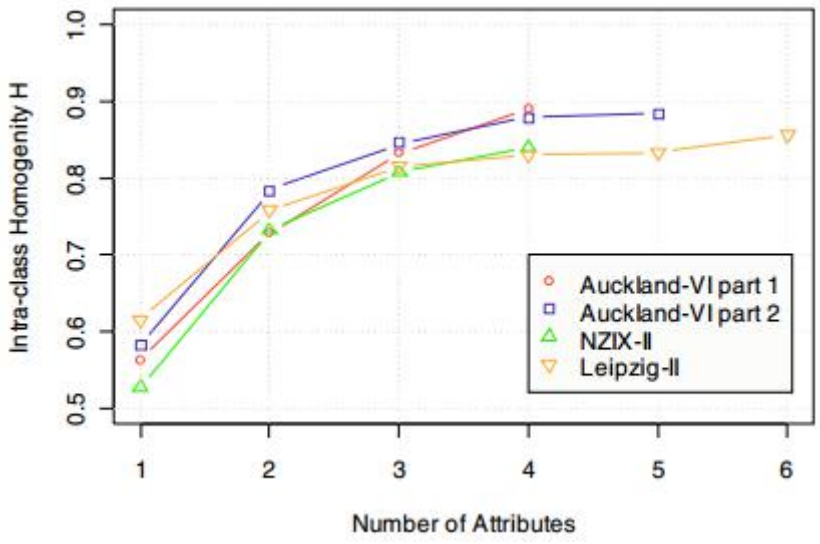

图 3：同质性取决于流量属性的数目和不同的轨迹

图 4 显示了每条轨迹的最佳集合已经选择的属性。y 轴是轨迹的一个特征进入最佳集合的百分比。虽然所有轨迹的最佳特征集合是不同的，但对一些特征有明显的趋势。数据包长度的统计数据似乎比到达间隔时间统计数据更好，持续时间和落后量也似乎是价值有限。

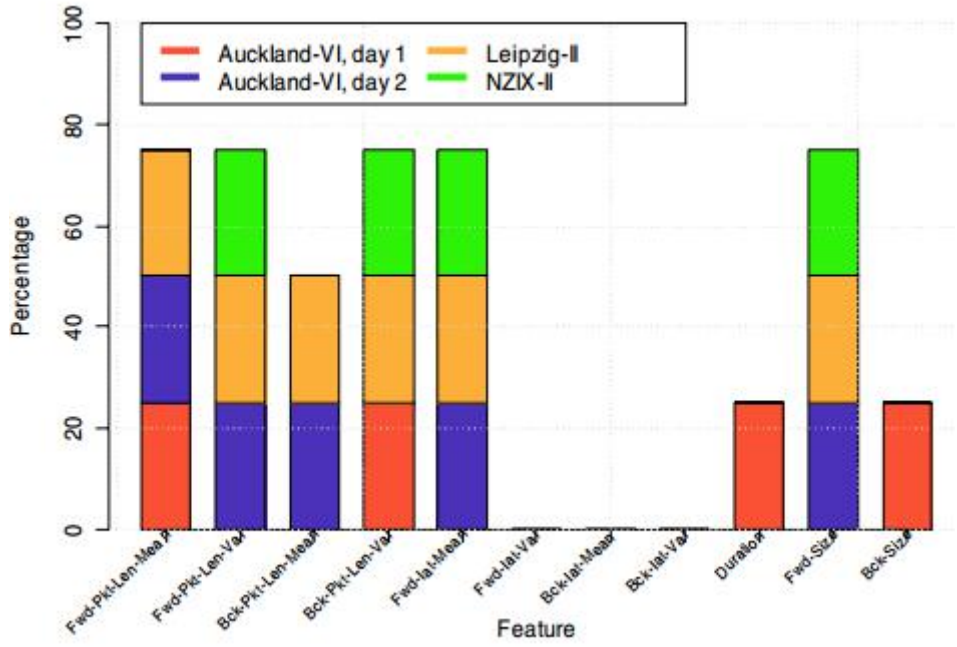

图 4：不同轨迹最佳特征集的入选特征

我们认为，之所以数据包长度优于到达间隔时间，是因为我们研究的所有应用都没有非常特征性的到达间隔时间分布。例如，如果我们选择了语音通信，这些应用程序有特征性的到达间隔时间（例如，每 20ms 一个数据包到达），那么我们会预测到达间隔时间更加有用。

游戏的流量，如半减期流量，被熟知有特征性的到达间隔时间。然而，这仅适用于在游戏中交流游戏状态信息时的流量。我们数据集中的大多数半减期流量实际上是由玩家从服务器查询信息导致的，如查询活跃玩家数等。到达间隔时间的一个潜在问题是，在路由器排队的数据包能改变它们的分布，特别是在拥堵的情况下。相比之下，如果数据包没有中间碎片或加密，长度通常是不变的。

我们也估计了不同属性对学习结果的影响。属性对一个特定类的影响被定义为，该类分布的交叉熵，关于一个类分类的全局分布。一个属性的总影响是在每个类中影响的类概率的加权平均。影响值的范围是 0（无影响）到 1（最大影响）。表 1 为跨不同轨迹的平均值（基于使用最佳属性集时的学习结果）。结果类似于图 4，数据包长度和体积的统计数据是最有影响的，而到达间隔时间和持续时间影响最小。

表 1：属性影响

Attribute	Influence
Forward-Pkt-Len-Var	1.0
Backward-Pkt-Len-Var	0.89
Backward-Bytes	0.84
Forward-Pkt-Len-Mean	0.77
Forward-Bytes	0.75
Backward-Pkt-Len-Mean	0.69
Duration	0.62
Forward-IAT-Mean	0.56

我们为每个不同的应用程序计算类的同质性。图 5 显示了跨不同轨迹的每个应用的同质性分布。每个应用的同质化值定义为其所有类同质化值的均值，在这些类中，该应用占有最大部分。该分布显示为箱线图。盒子的下端、中间和上端分别是分布的第一四分位数、中位数和第三四分位数。线延长到极值。

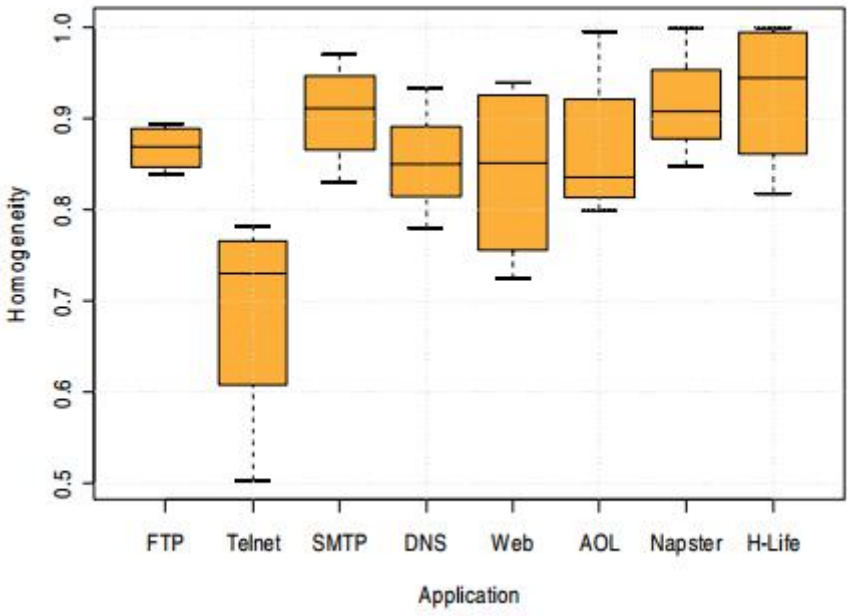

图 5：跨不同轨迹每个应用的平均同质化值

图中表明，一些应用的类别有同质性（例如半减期），但其他的几乎没有同质性（如远程登录，网页）。同质性越高就越有可能将一个应用程序从所有其他的程序中分离。一些应用，如半减期，能很好地从其他应用中分离出来，但其他的，如 FTP，似乎有非常类似于其他应用的特征。

图 6 显示了每个应用和所有轨迹“正确的”类别中流量的百分比（这通常称为精度）。为了计算精度，我们把每个类映射到控制该类的应用（该应用在这个类中占有流量的最大部分）。这个数字表示预期的分类精度。它表明，一些应用有非常高的精度，但是有一些问题，例如，对于 Napster 应用来说，有一条轨迹，它不控制任何类（因此精度为 0%）。然而，应该指出的是，中位数总是 80%或更高，对于某些应用来说，中位数接近 95%。跨所有跟踪文件的平均精度是 86.5%。

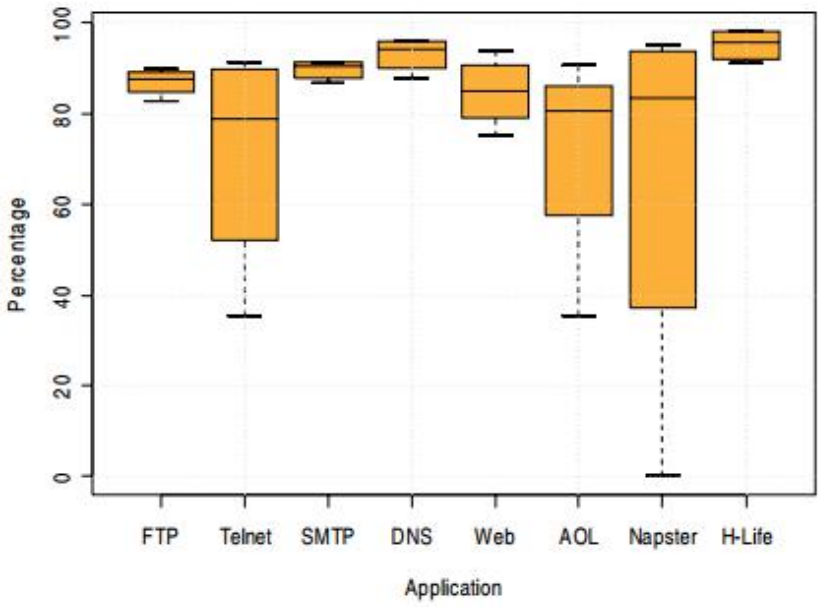

图 6：跨不同轨迹的每个应用的精度

虽然精度给出了分类正确流量的百分比，但是它不能度量哪些应用的流量很可能是误判。为了解决这个问题，我们也计算每个应用的假正率，它被定义为，误分类流量的数量除以分配给该应用的所有类中流量的总数。图 7 是每个应用假正率的百分比。FTP、远程登录和网页流量的假正率百分比最高。远程登录有和其他应用严重重叠的问题，而 FTP 和网页似乎有最多样化的流量特征，它们散布在许多类中。

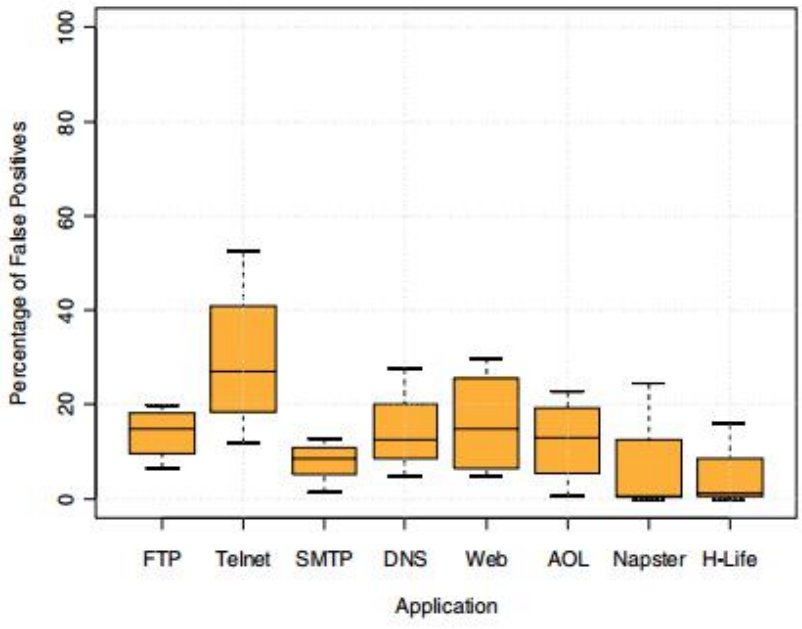

图 7：跨不同轨迹每个应用的假正率

为了可视化有最多样性特征的应用程序，图 8 展示了至少有一个流量的应用的类别百分比（应用在不同类中传播）。不足为奇的是，网页流量是最分布式应用（类似于 [13]中发现的结果），同时，游戏的流量是最不分布的。平均来说，认为类的总数是 100。

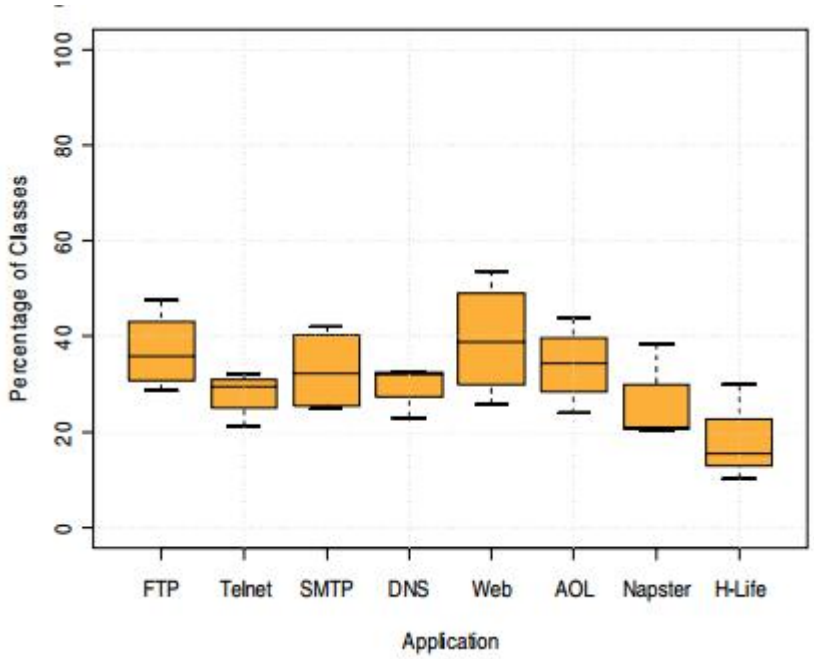

图 8：应用通过类在不同的轨迹中传播

虽然，主要的重点是，为了高分类精度获得不同应用间好的分离，最小化类别数目来提高学习、分类的速度以及最小化内存需求，这也是可取的。

5. 总结和展望

我们提出了基于机器学习的流量分类和基于统计流量特性的应用识别的新方法。我们使用特征选择技术寻找流量属性的最优集，并评估方法的效率。我们也量化了不同属性对学习的影响。研究结果表明，可以实现应用程序的分离，这取决于具体的应用。跨所有轨迹的平均精度是 86.5%。虽然一些应用似乎有更多的特征属性，能被很好地分离，其他一些混杂的更难被识别出。

我们计划用更多的流量和应用来评估方法。使用含有效载荷的数据包轨迹或流量数据会提高评估，其中，应用程序已经确定。我们用简单的贝叶斯分类器从事计算我们的算法，像在[16]中使用的一样。

清楚某些应用不能轻易从其他应用中分离的原因，研究新的、更好的流量属性来提高同质性，都是重要的。我们想用，如闲置时间、突发性和从有效载荷信息中以一种协议无关的方式计算的指标，这样的属性做测试。

所得分类器的精度和检索率以及分类器的性能需要评估。目前，我们还没在结果中调查流量取样的效果，尽管值得注意的是，任何类型数据集的选择已经代表抽样的某种形式。在高速网络中，数据包样本的使用是不可避免的，知道抽样误差如何影响流量属性是很重要的（见[24]）。另一个有趣的问题是，对可靠的识别来说，需要一个流量的多少数据包（在大多数情况下，流量必须尽可能快地被分类），随着时间的推移分类是如何稳定的（预测类应该只在网络应用改变时改变）。

大概我们的方法也能用于检测安全事件，如端口扫描或其他恶意流量，但目前，我们还没进入这个领域的研究。另一个有待研究的问题是，根据处理时间和内存消耗来量化性能，调查方法的精度和处理开销之间的权衡。

6. 参考文献

[1] IANA,http://www.iana.org/assignments/port-numbers(as of August 2005).
[2] Ports database,http://www.portsdb.org/(as of August 2005).
[3] Cisco IOS Documentation,“Network-Based Application Recognition and Distributed Network-Based Application Recognition”, http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/dtnbarad.htm (as of August 2005).
[4] S. Sen, O. Spatscheck, D. Wang, “Accurate, ScalableIn-Network Identification of P2P Traffic UsingApplication Signatures”, WWW 2004, New York,USA, May 2004.
[5] Reno, NV. "An Analysis of Flow Identification in QoS Systems", Poster at ACM SIGCSE 2003, Reno, USA,February 2003.
[6] K. Lan, J. Heidemann, “On the correlation of Internet flow characteristics”, Technical Report ISI-TR-574,USC/Information Sciences Institute, July, 2003.
[7] K. Claffy, H.-W. Braun, G. Polyzos, “Internet Traffic Profiling”, CAIDA, San Diego Supercomputer Center,http://www.caida.org/ outreach/papers/1994/itf/ , 1994.
[8] Tom M. Mitchell, “Machine Learning”, McGraw-Hill Education (ISE Editions), December 1997.

[9] Tom M. Mitchell, “Does Machine Learning Really Work?”, AI Magazine 18(3), pp. 11-20, 1997.
[10] J. Frank, “Machine Learning and Intrusion Detection:Current and Future Directions”, Proceedings of the National 17th Computer Security Conference, 1994.
[11] T. Dunnigan, G. Ostrouchov, “Flow Characterization for Intrusion Detection”, Oak Ridge National Laboratory, Technical Report,http://www.csm.ornl.gov/~ost/id/tm.ps, November 2000.
[12] M. Roughan, S. Sen, O. Spatscheck, N. Duffield,“Class-of-Service Mapping for QoS: A statistical signature-based approach to IP traffic classification,ACM SIGCOMM Internet Measurement Workshop 2004, Taormina, Sicily, Italy, 2004.
[13] A. McGregor, M. Hall, P. Lorier, J. Brunskill, “Flow Clustering Using Machine Learning Techniques”,Passive & Active Measurement Workshop 2004 (PAM 2004), France, April 19-20, 2004.
[14] A. Soule, K. Salamatian, N. Taft, R. Emilion, and K.Papagiannaki, “Flow Classification by Histograms or How to Go on Safari in the Internet”, In ACM Sigmetrics, New York, U.S.A., June, 2004.
[15] S. Zander, T.T.T. Nguyen, G. Armitage, ″Self-learning IP Traffic Classification based on Statistical Flow Characteristics″, Passive & Active Measurement Workshop (PAM) 2005, Boston, USA, March/April 2005.
[16] D. Zuev, A. Moore, “Traffic Classification using a Statistical Approach”, Passive & Active Measurement Workshop, Boston, U.S.A, March/April 2005.
[17] A. Moore, and K. Papagiannaki, “Toward the Accurate Identification of Network Applications”, Passive &Active Measurement Workshop, Boston, U.S.A.,March/April, 2005.
[18] NetMate, http://sourceforge.net/projects/netmate-meter/(as of August 2005).
[19] P. Cheeseman, J. Stutz, “Bayesian Classification(Autoclass): Theory and Results”, Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, USA, 1996.
[20] A. Dempster, N. Laird, D. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm, Journal of Royal Statistical Society, Series B, Vol. 30, No. 1, 1977.
[21] M. Hall, “Correlation-based Feature Selection for Machine Learning″, Ph.D diss., Waikato University,Department of Computer Science, Hamilton, NZ, 1998.
[22] Y. Yang and J. O. Pedersen, “A comparative study on feature selection in text categorization″, Proceedings of ICML’97, 14th International Conference on Machine Learning″, pp. 412-420, 1997.
[23] NLANR traces: http://pma.nlanr.net/Special/ (as of August 2005).
[24] N. Hohn, D. Veitch, “Inverting Sampled Traffic”,ACM/SIGCOMM Internet Measurement Conference,Miami, USA, November 2003.