预测所有端口上的 IPv4 服务

Liz Izhikevich
斯坦福大学

Renata Teixeira
Inria，巴黎

Zakir Durumeric
斯坦福大学

arXiv:2303.00895v1 [cs.NI] 2 Mar 2023

摘要

全互联网扫描通常用于了解互联网的拓扑结构和安全状况。然而，IPv4互联网扫描仅限于扫描部分服务——穷举扫描所有IPv4 服务的成本过高，而且现有的节省带宽框架都无法扫描所有端口上的 IPv4 地址。在这项工作中，我们介绍了 GPS，这是一个能够高效发现所有端口上的互联网服务的系统。GPS 运行一个预测框架，可从极小的样本量中学习，并且高度支持并行处理，因此能够快速发现所有 65K 端口上的服务与众多特征之间的模式。GPS 仅用 13 分钟便计算出服务预测（比以往工作快四个数量级），并且与穷举扫描相比，以 131× 更少的带宽和 204× 更高的精确率，在所有 端口上发现了 92.5% 的服务。GPS 首次证明，只要某个端口上至少有两个响应式 IP 地址可用于训练，就能够以切实可行的方式预测所有端口上的大多数服务。

CCS 概念

• **网络** → 网络结构； • **计算方法**→ 模型开发与分析； • **信息系统** →数据挖掘；

关键词

互联网扫描、预测、IPv4、测量

ACM 引用格式：

Liz Izhikevich、Renata Teixeira 和 Zakir Durumeric。2022。预测所有端口上的 IPv4 服务。载于*ACM SIGCOMM 2022*会议（*SIGCOMM '22*），*2022*年*8*月*22*日至*26*日，荷兰阿姆斯特丹。ACM，纽约，纽约州，美国，13页。https://doi.org/10.1145/3544216.3544249

1 引言

全互联网扫描使研究人员和网络运营商能够了解互联网在实际运行中的工作方式，并已用于研究网络拓扑 [16, 38], 、运营商决策 [20, 28] 以及安全漏洞 [15, 17]。全互联网扫描通过与一组指定端口上的服务建立网络连接来进行。遗憾的是，目前尚无研究能够分析所有端口上的整个 IPv4 服务空间，因为使用 ZMap [21] 以 1 Gbps 的速率扫描全部 37 亿个 IPv4 地址上的全部 65K 端口，需要 5.6 年——这一扫描速率可避免淹没目标网络。

特此授予许可，允许为个人或课堂使用而免费制作本作品的全部或部分数字副本或纸质副本，但前提是不得出于营利或商业利益制作或分发副本，并且副本首页须载有本声明及完整引文。对于本作品中由作者以外的其他方拥有的组成部分，必须尊重其版权。注明出处后进行摘要是允许的。若要以其他方式复制、重新出版、发布到服务器或重新分发到列表，则需要事先获得具体许可和/或支付费用。请通过 permissions@acm.org 申请许可。*SIGCOMM '22，2022*年*8*月*22*日至*26*日，荷兰阿姆斯特丹 © 2022 版权所有归所有者/作者所有。出版权授权给计算机协会。ACM ISBN 978-1-4503-9420-8/22/08...$15.00
https://doi.org/10.1145/3544216.3544249

因此，全互联网研究通常只扫描指定端口中相关的一部分（例如 23/TELNET 和 2323/TELNET），而 Censys [19] 和 Shodan [37] 等热门互联网服务"搜索引擎"也只能扫描数量最多的端口。

然而，近期研究表明，大多数互联网服务并不运行在指定端口上。仅扫描端口 80 会漏掉全部 IPv4 HTTP 服务的 97%（18亿个），而仅扫描端口 23 会漏掉 Telnet 服务的 95%（680万个） [25]。服务分布在非标准端口的长尾中 [25]：扫描数量最多的五千个端口的互联网服务搜索引擎（例如 Censys）会漏掉大多数互联网服务——据估计为 19亿 个（63%）。更复杂的是，非标准端口上的服务并不能由标准化端口上的服务准确代表：物联网设备和安全关键设备出现在非标准端口的19亿个服务中的概率最高可达五倍 [25]。不幸的是，僵尸网络已经开始攻击非标准端口上的易受攻击服务 [13]。此外，当研究人员和网络运营商需要在互联网服务的长尾中寻找"大海捞针"时，往往无法依赖子采样。例如，Marczak 等人依靠对选定端口进行完整的全互联网扫描，以识别间谍软件基础设施 [31] 以及网络间谍活动的客户端 [32]，而这些目标仅分布在几百台受感染服务器上。为了保护整个互联网，研究人员和运营商必须找出所有 IP 地址和 端口上的服务。

在本文中，我们介绍了 GPS，这是一种智能扫描系统，能够以可扩展且高效的方式，在所有 IPv4 地址和端口上查找服务。GPS 不依赖任何先验知识，而是首先收集一组初始"种子"数据。它利用大量应用层、传输层和网络层特征，以概率方式对服务存在性进行建模。GPS 使用该模型构建了一个分两个阶段进行的扫描方法：(1) 在所有主机上至少找到一个服务；(2) 利用已发现的服务，在同一主机上查找其余服务。GPS 的关键贡献是一种可并行化且准确的预测算法——通过计算特征与服务之间的简单条件概率——该算法只需极少的训练数据，即可在所有端口上并行发现模式并预测新服务。

GPS 使用比穷举扫描少 131 倍的带宽和高 204 倍的精确率，在全部 65K 端口上发现了 92% 的服务（其中响应式 IP 地址超过两个）。我们表明，随着服务越来越难以预测，发现新服务所需的带宽也会增加：在所有端口上发现 96% 的服务时，相比穷举扫描仅少使用 10 倍带宽。在仅扫描常用端口时，GPS 的算法所使用的带宽最多比当前最先进的基于机器学习的解决方案 [36] 少 28 倍。在单个 CPU 核心上，GPS 计算预测的速度是将相关工作扩展为预测所有端口上的服务时的五倍。在使用弹性云平台时，GPS 的可并行化算法可在 13 分钟内完成预测计算，比扩展相关工作快四个数量级。

GPS 是首个能够在 IPv4 地址空间的所有端口上找到大多数服务的实用系统。我们将发布

采用 Apache 2.0 许可证发布 GPS，让个体研究人员、网络运营商和安全公司能够减少所需带宽，以便发现、研究并保护互联网上大多数暴露在外、甚至可能已遭利用的服务。

2 相关工作

为降低扫描成本并发现更多服务，先前的工作提出了多种解决方案，用于预测在给定端口上应扫描哪些 IP 地址。先前的工作存在两个共同缺陷：（1）没有任何系统能够预测所有 65K 端口上的服务，或能扩展到这一规模；（2）现有解决方案需要大量难以承受的训练数据，由于搜索空间稀疏，收集这些数据将耗费数年。

分类器。 Sarabi 等人 [36] 将智能全互联网扫描视为一个分类问题——每个端口值对应一个类别——并使用 XGBoost 分类器 [18]预测某个 IP 地址是否会在 20 个常用端口之一上响应。他们的研究发现，服务最有力的预测因素是同一主机上是否存在其他服务。为了发现常用端口之间的模式，他们的系统按顺序为每个端口训练一个模型，并使用每个模型的输出训练下一个模型。遗憾的是，由于两个原因，他们的方法无法扩展到所有 65K 端口。首先，每个端口至少需要 1000 万个服务实例来训练模型，而 99.99% 的端口都无法提供这么多数据（许多端口上的服务实例数量根本达不到这一规模）。其次，每个端口都必须单独训练一个模型，因此计算时间高得难以承受：在 NVIDIA GeForce GTX 1080 Ti GPU 上，每个端口的训练和预测过程约需 70 秒，而覆盖所有 65K 端口则约需 53 天计算时间。由于模型是按顺序训练的，这一计算过程无法并行化。我们在第 6.4 节比较 XGBoost 扫描器与 GPS 的性能，并表明 GPS 的平均准确度高于 XGBoost，能够预测所有端口上的服务，且计算服务预测所需的时间少了四个数量级。

目标生成算法。 先前的工作 [22, 23, 35] 使用目标生成算法（TGAs），预测 IPv6 地址上运行的服务。TGAs 会学习已知 IP 地址的结构，并预测结构相似且可能运行服务的地址。然而，TGAs 无法发现遍布所有 65K 端口的模式，而且每个端口都需要构建并训练一个新模型。TGAs 还完全依赖子网相关性，而我们表明，对于非常用端口上的主机，其预测能力要低几个数量级（第 4 节）。最重要的是，TGAs 在计算上无法扩展：若要在 90% 的端口上获得有效训练模型所需的最少 IP 地址（即 1,000 个 IP [22]），由于大多数非常用服务都十分稀疏，每个端口至少需要随机探测 25% 的地址空间；使用 ZMap [21] 以 1Gbps 速率收集这些数据将需要一年多时间。

我们通过修改 En- tropy/IP [22] 和 EIP [23]，使其预测 IPv4 地址（每次预测一个 IPv4 八位组，而非一个 IPv6 半字节），以验证 TGAs 是否能够预测 IPv4 地址上高密度端口的服务。我们从 Censys 的通用互联网数据集中随机抽取 1,000 个地址，用于训练每个端口对应的模型；该数据集包含覆盖 2K 个端口的 100% IPv4 扫描。每个模型针对每个端口预测 1M 个候选地址（数量级比在 90% 的端口上有响应的 IP 数量多一个数量级）。合并所有候选地址后，Entropy/IP 和 EIP 只能找到 Censys 数据集中 19% 的服务。

推荐系统。 专有推荐系统已成功为数百万用户推荐数百万件商品 [24, 30, 33]。我们应用一种广受欢迎的开源混合推荐器 [26]，为 IP 地址推荐候选端口，但发现其在预测服务方面并不成功（附录 A）。

3 GPS 目标

GPS 的目标是在所有端口上尽可能多地发现服务。这不同于以往旨在最大化所发现服务"服务比例"的互联网扫描系统 [36]（公式 1）：即系统发现的服务数量，相对于在"真实值"集合中发现的 IP 地址数量（即端口 p上真实服务数量的基准，通过 100% 扫描获得）。该指标偏向于在服务更为密集的常用端口上发现服务（例如，全部 65K 端口上的所有服务中，有 5% 位于排名前 10 的端口），并且不鼓励在研究不足且更可能存在漏洞的非常用端口上发现服务 [25]。

$$\text{Fraction of Services} = \frac{\#(IP, p) \text{ Found by System}}{\#(IP, p) \text{ in Ground Truth}} \tag{1}$$

为解决这一问题，我们引入一项额外的归一化服务 指标（公式 2）。对于给定端口 p，该指标根据在"真实值"集合中对端口 p（即 IP_p）作出响应的 IP 数量，对服务权重进行归一化。这样一来，在不常见端口上发现所有服务，与在常见端口上发现所有服务具有同等权重。使用这一新指标，GPS 的目标（公式 3）是在所有端口的集合（|P|）中，最大化已发现归一化服务的比例，同时将探测包数量（即带宽）限制在常数 c_1 以内。

$$\text{Normalized Services} = \frac{\sum_{p \in \mathcal{P}} \frac{\#IP_p \text{ Found by System}}{\#IP_p \text{ in Ground Truth}}}{|\mathcal{P}|} \tag{2}$$

$$\max \text{Normalized Services}(bandwidth) \quad bandwidth < c_1 \tag{3}$$

通过限制带宽，GPS 会优化为仅扫描最容易预测的服务，以最大化已发现归一化服务的总比例。可用带宽越大，GPS 发现的服务就越多。

构建可部署系统时，还需满足另外两个约束：

(1)计算可扩展性： 预测所有端口和 IP 地址上的服务涉及庞大的搜索空间，但需要一种能在有限墙钟时间内运行的解决方案。随着互联网服务不断变化，这一点尤为重要 [34]。例如，我们在 6 月 27 日和 10 月 17 日，分别扫描 IPv4 地址空间中相同的 0.1% 以及 K 个端口。在这段 9 天的时间里，% 的归一化服务和 % 的所有服务都消失了。因此，如果预测服务耗时过长，模型用于学习模式的初始服务集，

以及模型的预测结果，很可能会过时。

(2) **缺乏先验知识：** 智能互联网扫描并非一项经典预测任务，因为这类任务一开始就具备有用的特征供训练使用。预测特征位于起初未知的 IP 地址和端口上。由于 GPS 在开始时并没有关于互联网服务位置的先验信息，因此必须设计高效的扫描策略，逐步发现有关主机的信息，之后才能利用这些信息预测其他服务。

伦理。 任何开源扫描技术都可能被研究人员用于监测和保护互联网服务，也可能被攻击者用于发现漏洞。我们设计 GPS 时，考虑到网络运营商可以轻松阻止它（第 5.5 节），这会抑制攻击者使用 GPS，同时让研究人员能够了解互联网的状况并保护互联网。在执行扫描和构建 GPS 时，我们遵循 Durumeric 等人 [21] 概述的互联网公民行为准则。我们还强调，GPS 可减少扫描大量端口上的服务时发送的流量（第 6 节），从而减轻对目标网络的影响。我们希望研究人员和公司都考虑使用 GPS，以减少互联网上扫描探针的总数。

4 识别预测特征

乍看之下，要在 37 亿个地址和 65K 个端口上查找 IPv4 服务——搜索空间近乎 2_{48} ——似乎难以实现。然而，先前的研究揭示了几种预测热门 服务位置的方法。我们首先探究这些预测模式能否用于预测所有 65K 个端口上的服务。研究结果构建了 GPS 将依赖的三类预测特征的基础集合：网络层、传输层和应用层。

端口使用情况存在相关性（传输层）。 Bano 等人发现，在最热门的八个端口中，主机上某个端口的存在情况可用于预测其他端口的存在情况 [14]。我们在2021年3月对一个占比为 1% 的随机 IPv4 样本扫描了全部 65K 端口，发现这一规律适用于绝大多数端口：对于每个端口，至少有 25% 的主机也会在同一个第二端口上响应。因此，服务所使用的端口可用于预测同一主机上的其他开放端口。

不同类型的主机更有可能运行特定的服务（应用层）。 相关工作 [25, 27] 表明，物联网设备和路由器厂商通常会制造让特定端口保持开放的设备，以提供网络访问。因此，表明制造商或操作系统的应用层数据（例如 TLS 证书）很可能可用于预测同一主机上的其他服务。使用同一次扫描，我们手动调查了所有主机中最常见的前 3K 个 HTTP 标头值、SSH 横幅和 TLS 证书。我们发现，在大多数端口上，物联网设备和路由器是最常见的主机类型。在所有端口上，由制造商生产、因而端口存在情况可预测的设备主导服务，这促使我们进一步研究预测非常用端口上的服务。能够识别主机制造商（例如 TLS 证书的组织、主题名称或颁发者、SSH 横幅、PPTP 厂商等）、操作系统（例如 HTTP 服务器、SSH 横幅、客户终端管理协议标头、MySQL 服务器版本等）、用途（例如 HTTP HTML 标题、VNC 桌面名称等）以及所有者（例如 SSH密钥、TLS 证书等）的应用层特征，可以预测服务存在。

互联网服务更有可能在同一网络中共同出现（网络层）。 Murdock 等人和 Foremski 等人发现，给定网络中的主机更有可能开放相同的端口 [22, 35]。我们使用 Censys 的通用数据集，该数据集包含对超过 2K 个端口的 100% IPv4 扫描 [11], , 发现 81% 的所有服务至少有两次在同一 /16 子网内的同一端口上出现。这一结果表明，网络可以预测服务是否存在。然而，随着端口变得越来越不常用，在同一 /16 子网内发现两个服务在同一端口上响应的概率会低至 0.02%。因此，虽然网络模式有助于发现开放常用端口的主机，但仅依靠网络模式无法发现使用非常用端口的服务。

在第 5.2 节中，GPS 使用传输层、应用层和网络层特征来预测服务是否存在。

5 系统架构

本节将介绍 GPS，这是一个能够高效发现所有端口上的互联网服务的系统。GPS 不依赖任何先验知识，而是通过四个阶段逐步启动自身并全面发现服务。首先，GPS 通过随机抽样扫描收集一组可供学习的互联网主机和服务，称为“种子集”。其次，GPS 基于种子集中的所有服务，创建一个概率模型，用于识别最具预测性的特征值。第三，由于 GPS 的种子集仅涵盖部分主机，因此 GPS 使用其概率模型，在每台可响应的 IPv4 主机上至少找到一项服务。找到主机上的首个服务并非易事，且会消耗大量带宽，但这对预测其余服务至关重要。第四，在每台可响应的主机上至少发现一项服务后，GPS 会预测其余服务。

GPS 的关键洞见是，使用一种简单且可并行化的计算方法来预测服务，且不依赖大型训练集。尽管 GPS 算法的计算开销很大，我们仍提供了一种实现方案，将无服务器计算的弹性资源与 GPS 的并行化相结合，从而将 GPS 的实际预测时间缩短数个数量级。

5.1 构建种子集

GPS 对互联网主机一无所知。为了学习模式并发现服务，GPS 必须收集，或使用现有的（例如 LZR 数据集 [3]）IPv4 主机均匀随机样本，并扫描每台主机的所有 65K 端口（即“种子集”）。种子扫描的规模直接影响预测质量，也是确保 GPS 学会所有预测模式的主要方式。我们在附录 D.2 中展示了确定种子扫描规模时的带宽/覆盖率权衡：种子大小越大，发现的服务越多，但所需带宽也显著增加。重要的是，与需要大样本量（例如 25% 的 IPv4 样本）进行训练的以往工作（第 2 节）不同，GPS 仅凭 0.1% 的 IPv4 样本，就能预测大多数端口上的服务。GPS 允许用户根据带宽限制，通过输入参数指定所需的种子扫描规模（公式 3）。

应用层或 网络层特征	# 中的唯一值 Censys 基准真值
协议	56
TLS 证书：哈希	30.1M
TLS 证书：组织	1.1M
TLS 证书：主题名称	27.9M
HTTP：HTML 标题	5.9M
HTTP：正文哈希	50.8M
HTTP：服务器	480K
HTTP：头部	22K
SSH：主机密钥	14.3M
SSH：横幅	177K
VNC：桌面名称	4.5K
简单邮件传输协议：横幅	2.9M
文件传输协议：横幅	1.5M
互联网邮件访问协议：横幅	144K
邮局协议第3版：横幅	390K
客户终端管理协议：头部	10
客户终端管理协议：正文哈希	11
Telnet：横幅	219K
点对点隧道协议：供应商	390K
MySQL：服务器版本	5.7K
Memcached：服务器版本	129
Microsoft SQL Server：服务器版本	381
智能平台管理接口： 横幅	116
IP 的 /16 子网	37.3K
IP 的自治系 统编号	67.7K

表 1：GPS 特征—GPS 的特征涵盖了 Censys 上所有具有可用横幅的 TCP 协议（即 15 种唯一协议）。GPS 仅使用对服务存在最具预测性的网络特征；筛选过程见附录 C。维度（即唯一值的数量）根据第 6.1 节所述的 Censys 真实数据集计算得出。

5.2 识别预测模式

GPS 的第二阶段完全是计算过程：构建概率模型，以识别最能预测服务存在的特征模式。GPS 在流程的第三和第四阶段使用该概率模型来扫描新服务（第 5.3 节和第 5.4 节）。

特征选择。为识别对服务存在最具预测性的特征值，GPS 使用种子集提取第 4 节介绍的三类特征——应用、传输和网络。我们的评估中，GPS 共使用 25 个唯一的应用层和网络层特征（表 1）。GPS 的特征涵盖了 Censys 上所有具有可用横幅的 TCP 协议（即 15 种唯一协议）。因此，GPS 兼顾了流行协议（如 HTTP）和较不流行的协议（如 VNC）都可能包含可用于预测服务存在的信息。GPS 的设计便于用户轻松纳入或移除任何特征候选项。

交互建模。 GPS 会独立建模三个主要特征类别之间不同的交互 。这考虑到了三个特征类别的任意组合都可能有助于预测互联网服务。例如，当一台主机出现在多个子网中时，了解其网络 (Net_{IP}) 的预测能力可能较弱；而当某项服务只存在于一个子网中时，预测能力可能较强。

例如，安卓电视可能出现在许多子网中；而 Freebox 设备只出现在 Free 网络 [4] 中。具体来说，GPS 对以下内容进行建模：

1. 传输层对应关系：已知主机有 $Port_b$ 开放，预测该主机将有 $Port_a$ 开放的概率（表达式 4）。

$$\mathbb{P}(Port_a|Port_b) \tag{4}$$

2. 传输层与应用层对应关系：已知主机有 $Port_b$ 开放，且 $Port_b$ 上的服务包含特定应用层特征值，预测该主机将有 $Port_a$ 开放的概率（表达式 5）。

$$\mathbb{P}(Port_a|(Port_b, App_{Port_b})) \tag{5}$$

3. 传输层与网络对应关系：已知主机有 $Port_b$ 开放，且在 $Port_b$ 上响应的主机位于网络 Net_{IP} 中，预测该主机将有 $Port_a$ 开放的概率（表达式 6）。

$$\mathbb{P}(Port_a|(Port_b, Net_{IP})) \tag{6}$$

4. 传输层、应用层与网络对应关系：已知主机有 $Port_b$开放，且 $Port_b$ 上的服务包含特定应用层特征值，在 $Port_b$ 上响应的主机位于网络 Net_{IP} 中，预测该主机将有 $Port_a$ 开放的概率（表达式 7）。

$$\mathbb{P}(Port_a|(Port_b, App_{Port_b}, Net_{IP})) \tag{7}$$

请注意，所有条件概率都依赖于 $Port_b$ 已被发现，因此可以使用相关应用层和/或传输层特征值来预测 $Port_a$。不过，并非所有主机都会在多个端口上响应，也并非所有主机都事先拥有可用信息（例如已知服务）。下一节将说明 GPS 如何利用其概率模型预测服务，无论主机响应的端口数量是多少，也无论可用的主机先验信息有多少。

计算可扩展性。 我们注意到，GPS 用于建模特征交互的方法计算开销很大；GPS 必须计算种子集中出现的所有可能的特征组合。此外，尽管端口特征（$Port_a$）的维度为 65K，应用层特征的维度却可能差异很大（表 1），因此可能产生数十亿种特征值组合。尽管如此，GPS 的方法在所有 65K 端口上都具有计算可扩展性，因为条件概率的计算可以在所有 65K 端口间并行执行。我们将在第 5.5 节介绍概率模型的实现，以及如何利用无服务器计算大幅缩短墙钟时间，并使用矩阵运算扩展计算规模。我们将在第 6.5 节正式评估 GPS 的计算复杂度。

5.3 预测首个服务

预测主机上的首个服务，与预测该主机上的其余服务，必须采用不同的方法。预测主机上的首个服务时，只能利用网络层特征（即 IP 地址所属的子网）来预测服务存在。相反，一旦已知至少一个服务，就可以利用该服务提供的信息来预测同一主机上的其他服务（例如端口相关性）。由于种子集只包含

关于所有主机中一部分（例如 1%）的信息，GPS 必须先在每台可响应的 IPv4 主机上发现至少一个服务，才能利用它进一步预测同一主机上的其他服务。

扫描步长。 如第 4 节所述，服务更可能在子网中共同出现。因此，给定种子集中的一个有响应的服务（$IP,\ p$），GPS 必须在端口 p 上探测 IP 周围的 IP 地址（即其所在网络中的地址），以最大限度地提高发现其他服务的机会。

GPS 在构建高效的扫描策略时面临权衡：种子集中有响应服务所在的子网应扫描得多么彻底？在某个端口上扫描 100% 的 IPv4 地址空间，能提高发现所有在该端口上有响应的主机的可能性，但需要更多带宽，也会加大对目标网络的影响。例如，在种子集中 IP IP 所在的 /24 子网上扫描端口 p，所需带宽要少得多，但可能会漏掉该子网之外同样在端口 p 上有响应的主机。用户的带宽限制决定了 GPS 应采用多大的“扫描步长”（即穷举扫描某个端口的子网大小）；该参数由用户指定。我们在第 6.2 节中正式评估带宽/覆盖率权衡，以帮助用户确定应采用的扫描步长。

依赖对特定网络和端口进行选择性随机探测，虽然会消耗大量带宽，但在初期不可避免。因此，GPS 只会在绝对必要时才使用这种方法：对每台主机，只 预测必须首先找到的服务，以便预测其余所有服务。

选择预测能力最强的服务。 GPS 优先寻找数量最少且最能提供信息的服务集，以便预测其余所有服务。例如，对于一台主机，如果它在端口 80 上返回通用 HTTP 页面，而在端口 222 上返回揭示厂商信息的横幅，那么发现该横幅很可能比发现 HTTP/80 服务更有助于预测（大多数 HTTP/80 服务不会在 SSH/222 上响应，但大多数 SSH/222 服务会在 HTTP/80 上响应）。GPS 使用其概率模型（公式 4–7），计算种子集中的每台主机上哪个服务最能预测该主机的所有其余服务。如果一台主机在种子集中只有一个端口会响应，那么这个唯一的服务就是必须预测的第一个也是唯一一个服务。

GPS 执行以下算法，以确定要扫描哪些子网和端口，从而优先找到预测能力最强的服务的最小集合：

(1) 对于种子集中仅有一个端口响应的所有主机，保存该服务的 $(Port_a,\ Net_{IP})$。(2) 对于种子集中有多个端口响应的所有主机，计算全部四个条件概率（公式 4–7）。对于种子集中的每个 $(IP,\ Port_a)$，确定能使 IP($Port_a$) 达到最大的 $Port_b$，并保存 $(Port_b,\ Net_{IP})$。(3) 汇总所有主机，将所需的 $(Port,\ Net_{IP})$ 元组归为一组，并统计它们在种子扫描中有助于预测的不同服务数量（即最大覆盖率）。(4) 根据最大覆盖率对 $(Port,\ Net_{IP})$ 进行排序。

先验扫描列表。 GPS算法输出一个“先验扫描列表”：由唯一元组组成的有序列表，每个元组包含一个（端口、大小为扫描步长的子网）对（例如，（80，1.1.0.0/16））， 并按最大覆盖率排序。先验扫描列表使GPS能够并行收集所有端口上最具预测性的服务。请注意，并非所有端口和子网都会被扫描：只有那些能以最高概率发现所有剩余服务的（端口、子网）元组才会被扫描。扫描先验列表使GPS能够找到每台主机上最具预测性的服务，随后便可利用该服务预测每台主机上的其他服务。

5.4 预测其他服务

GPS 在每台主机上至少找到一个服务后，会从每个已发现的服务中提取三类特征——应用、传输和网络特征，以预测同一主机上的其他新服务。GPS 使用现有的概率模型（公式 4–7）生成“预测能力最强的特征”列表，并利用该列表预测其他服务。

GPS 用户 使用以下算法来预测其他服务：

（1）对于种子集中的每项服务，确定可使 IP($Port_a$) 达到最大值的特征元组（例如（$Port_b$、App_{Port_b}））。将该特征元组和预测端口 $Port_a$ 保存到“最具预测性的特征值”列表中。为考虑出现在随机端口上且最大 IP($Port_a$) 较低的服务，丢弃所有低于 0.00001 的概率；这大致相当于随机探测大多数端口时的命中率。（2）对于在 先验扫描中发现的每项响应服务，提取可用的特征值（例如 TLS 证书、SSH 主机密钥）。（3）对于在先验扫描中的每项响应服务，以及其所有特征值，如果该特征出现在“最具预测性的特征值”列表中（即 $Port_b$、App_{Port_b}），则将预测端口 $Port_a$ 和主机的 IP 保存到“预测”列表中。

GPS算法输出一个“预测列表”：一个按顺序排列的唯一 IP 地址和端口列表，用于扫描。步骤 1 对 GPS 算法至关重要；通过使用种子扫描中的每项服务来构建“最具预测性特征”列表，GPS 的预测算法保证，种子集中曾见过且“可预测”的每项服务，都会根据最有可能找到该服务的特征值模式进行预测。

5.5 实现

我们介绍 GPS 的一种实现，它充分利用其可并行化的算法，大幅缩短预测服务所需的墙钟时间。

为了进行全互联网扫描（例如，收集种子扫描、扫描先前的服务、扫描预测服务），GPS 将三种现有工具串联起来执行扫描：ZMap [21] + LZR [25] + ZGrab [12]。ZMap 是一种无状态的第 4 层扫描器（即“syn-scanner”），在 GPS 流水线中发起与互联网主机的 TCP 连接。由于 ZMap 具有独特的指纹（IP 标识 = 54321），GPS 使用 ZMap 也使其很容易被网络运营商屏蔽。随后由 LZR 接管 TCP 连接，过滤掉中间盒，并高效地识别服务指纹（扫描未分配端口时必不可少的一步）。对于指纹识别结果表明正在运行的确是真实服务的情况，GPS 会提供选项，让 LZR 将连接信息转发给 ZGrab；随后 ZGrab 可以完成完整的第 7 层握手，以收集更多应用层特征。

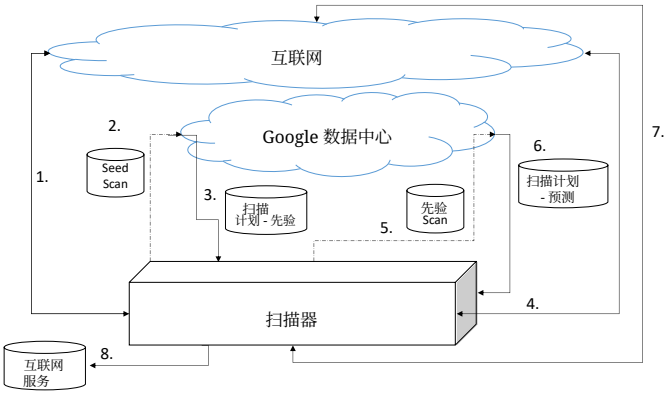


图 1：GPS实现——GPS的可并行化预测算法利用 Google BigQuery 的无服务器平台，在高度可并行化的执行环境中计算预测结果。

实现 GPS 用于识别预测模式的算法（第 5.2 节）面临一项挑战：找出特征与端口的所有两两组合——尽管这一过程可以并行化——但计算和内存开销都很大。因此，为了尽可能缩短墙钟时间，GPS 可以直接受益于高度可并行化的执行环境；理想情况下，系统的任何用户都可以使用这种环境。GPS 可以部署在任何服务器基础设施上，但我们发现，Google BigQuery 这种无服务器数据库服务能够对拍字节级数据进行可扩展分析 [1]，非常适合这项任务。使用数据库查询语言实现 GPS 的算法（例如计算条件概率、识别最具预测性的特征）是自然而然的选择，因为这些算法主要依赖于读取数据、汇总数据，以及基于共享数据字段进行联接。我们在第 6.5 节中正式评估了这一实现，并展示了在有无高度可并行化环境的情况下，GPS 分别比先前的工作快 5870× 和 5× 。

我们在图 1 中展示了 GPS 与 BigQuery 的交互方式；GPS 将 BigQuery 用作计算引擎，并不依赖任何现有的 BigQuery 数据。种子扫描完成后，GPS 会将扫描结果上传到 BigQuery。GPS 在 BigQuery 中通过以下方式提取服务特征：（1）从上传的扫描结果中选择相应字段；（2）对 IP地址 执行操作，以提取主机子网；或（3）与提供该特征的数据库（例如自治系统编号）进行联接。为高效计算条件概率，GPS 使用 BigQuery 的 SQL 语言，为每个特征和端口计算成对共现矩阵；这需要将数据集与自身联接，以找出特征的所有成对组合，并汇总相同的特征值模式和目标端口，从而计算条件概率。GPS 还使用 BigQuery 的 SQL 语言实现预测首个服务的策略（第 5.3 节），方法是将种子扫描与计算得到的共现矩阵进行联接。

计算出预测首个服务的策略后，GPS 会将该策略下载到其扫描主机上。扫描主机会对（端口、子网）元组进行穷举扫描，并将扫描结果上传到 BigQuery，BigQuery 随后再次提取主机特征。GPS 使用 BigQuery 的 SQL 语言创建最具预测性的特征列表（第 5.4 节）：将先验扫描与计算得到的共现矩阵进行连接，并将所有剩余服务的预测策略（即 IP 和端口列表）下载到扫描主机上。随后，扫描主机会扫描所有剩余的预测服务，并将结果保存在本地数据库中。我们已根据 Apache 2.0 许可证在 https://github。com/stanford-esrg/gps 上发布 GPS 的实现。

6 评估

本节评估 GPS 在所有端口上发现服务的能力。首先，我们评估 GPS 在带宽限制可变的情况下发现最多服务的能力。我们表明，随着服务越来越难以预测，发现服务所需的带宽也会增加：对于所有端口上超过两个响应式 IP 地址的服务，GPS 能发现其中 92%，且所需带宽比穷举扫描少 131×；但要发现所有端口上 96% 的服务，GPS 仅节省 10× 的带宽。其次，我们评估 GPS 的精确率。在尽量减少互联网扫描对可用互联网资源影响的情况下，GPS 的精确率比穷举探测高两个数量级。第三，在扫描常用端口时，我们将 GPS 的准确率和带宽与 XGBoost 扫描器 [36] 进行比较。GPS 在实现相同服务覆盖率的情况下，最多可节省 28 倍带宽，平均可节省 2.3 倍。第四，我们评估预测服务的计算复杂度，并表明与 XGBoost 扫描器相比，GPS 使用单核时计算预测的速度快 5 倍，使用 BigQuery 时则快四个数量级。最后，我们利用 GPS 的预测结果，找出最能预测服务存在的特征值。

6.1 方法

要正确评估 GPS，需要找到合适的真实值数据集，并对 GPS 进行适当配置。

近似真实值。要找到用于评估 GPS 的“真实值”数据集颇具挑战：目前没有方法能够高效地以 100% 覆盖率扫描全部 65K 端口（这正是需要 GPS 的原因）。因此，我们使用两个数据集评估 GPS。首先，我们使用 Censys 的通用数据集，该数据集包含 2021 年 7 月 26 日采集的、覆盖最热门 2K 端口 100% 的 IPv4 扫描 [11]，是目前公开可用的、以 100% 覆盖率扫描端口数量最多的数据集。Censys 表示，按照其扫描带宽，穷举扫描全部 65K 端口需要 196 天。

Censys 仅针对最热门端口。因此，我们还使用 LZR 扫描器 [25]，将 2021 年 4 月对所有 65K 端口进行的 IPv4 地址空间 1% 随机扫描结果作为基准评估 GPS；该扫描耗时整整 30 天。尽管 LZR 数据集涵盖所有端口，但每个端口可用的样本数量较少，因此可能无法捕捉少量主机所呈现的模式。我们按照附录 B 中所述的步骤，从两个数据集中筛选真实服务。

为了为每个数据集创建种子扫描和测试集，我们随机将每个 IP 地址及其对应的服务分配到种子集或测试集中。因此，Censys 的 2% 种子集对应 98% 的测试集，而 LZR 的 0.5% 种子集对应 0.5% 的测试集。尽管我们花整整一个月扫描了 IPv4 地址空间中 1% 的范围，覆盖 65K 端口，但大多数端口上的响应服务仍然很稀少。由于我们不期望 GPS 学习并预测没有

训练数据的端口上的服务，因此我们会筛选 LZR 种子集和测试集，仅纳入响应 IP 地址超过两个的端口。使用 0.5% 的种子集时，筛选后剩余 13,162 个端口。

参数调优。 GPS 要求将种子和步长指定为输入参数。我们评估了种子和扫描步长对 GPS 性能的影响，发现较小的步长会提高 GPS 的精确率，但会降低召回率（附录 D）。这种权衡的原因是，正如第 5.3 节所述，扫描较小的先验端口子网所需的带宽要少得多，但可能会漏掉子网之外、可能包含有用特征值的主机。增大种子大小会提高发现的归一化服务的比例，因为较大的种子大小更有可能遇到主导非常用端口的非常用模式。不过，种子大小不会显著影响发现的整体服务比例。我们在每项实验中均说明所选的种子大小和扫描步长。

特征。 我们指定 GPS 使用 25 个特征（表 1).

6.2 GPS 在所有端口上的覆盖率

GPS 的目标是最大化在所有 端口上发现的服务数量。为了量化覆盖率，我们采用两项指标：已发现全部服务的比例（第 3 节中的公式 1）和已发现归一化服务的比例（第 3 节中的公式 2）；后者对常用端口和不常用端口上的服务赋予相同权重。我们在 Censys（种子集占 2%）和 LZR（种子集占 0.5%）这两个数据集上计算这两项指标。作为参考，我们还绘制了：（1）预言机预测器所用的带宽，该预测器准确知道要探测哪些服务（即准确率为 100%）；（2）穷举探测能够最大化服务发现的最小端口子集时，发现的服务比例随带宽的变化情况（即端口 80、（80、443）、（80、443、7547）等）。与穷举扫描所有端口相比，这一“最优端口顺序”探测基准能更准确地估算出，为了发现最大比例的服务，至少需要穷举探测多少个端口。例如，穷举扫描最热门的 10 个端口（即全部端口的 0.015%），是能够发现全部服务中 5% 的最小端口集合。我们以 100% 扫描为单位报告带宽使用量（即 37 亿个数据包），以便轻松比较 GPS 性能与穷举扫描。

GPS 找到的服务数量取决于带宽预算（图2）。在针对跨越 2K 个端口的 100% IPv4 扫描（Censys）进行评估时，GPS 最初仅使用最优端口顺序探测所需带宽的 21×，就能找到 94% 的服务。然而，由于 GPS 按可预测性降序扫描服务，快速预测服务会变得更加耗费带宽：在跨越 2K 个端口的情况下，找到 98.2% 的服务仅比最优端口顺序探测节省 7.6× 的带宽。Censys 表示，他们每天探测相当于进行 572 次 100% 扫描的内容，以整理其数据集；GPS 使用 3×10^{11} 个探测包，即比 Censys 少 6.3× 个探测包，就能找到跨越 2K 个端口的 98.2% 服务。

在寻找归一化服务时，带宽与覆盖率之间的权衡更加突出：使用比最优端口顺序探测少 100× 的带宽，GPS 可在 2K 个端口上找到 46% 的归一化服务；但在找到 67% 的归一化服务时，节省的带宽仅为 50%。在针对所有端口的 1% IPv4 扫描进行评估时（响应 IP 地址多于 2 个），GPS 找到了 92.5%，并且分别以比最优端口顺序探测少 6× 和少 2× 的带宽找到所有服务的 95%。GPS 在针对所有端口进行评估时性能下降，可能是因为种子大小缩小了 4 倍；使用覆盖 2K 个端口的 0.5% 训练种子集时，GPS 的性能几乎完全相同（附录 D.2）。

我们在第 7 节中量化了 GPS 在预测服务时面临的根本局限性。

6.3 GPS 在所有端口上的精确率

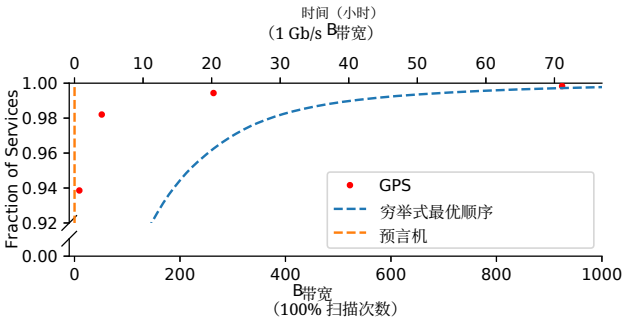
为了最大限度地降低全互联网扫描对目标网络的影响，我们表明，GPS 的精确率比随机探测高出两个数量级以上。GPS 首先扫描最具可预测性的服务（第 5.4 节）。因此，在扫描计划开始阶段寻找最具可预测性的服务时，GPS 所需带宽更少、精确率更高，并且能够最大限度地缩短发现时间。我们使用 Censys 数据集，以 1% 的种子大小针对 100% 扫描进行评估。为了最大化 GPS 的精确率，我们将 GPS 的扫描步长设置得较小（/20）。我们在图 3 中绘制了 GPS 随着持续找到服务而变化的精确率；作为比较，还展示了按照优先发现最多服务的最优顺序，对所有端口进行穷举探测时的精确率。

GPS 能以 36% 的精确率找到全部服务中最先出现的 1%，其精确率比穷举探测高一个数量级。GPS 最多能找到 94% 的全部服务和 46% 的归一化服务，而且精确率始终比穷举探测高一个数量级以上。例如，GPS 找到第 94 百分位的服务时，精确率比穷举探测高出 204×。请注意，随着 GPS 按可预测性从高到低不断用尽其预测结果，其精确率会随时间推移而下降。用尽所有预测结果后，可以选择将 GPS 配置为随机探测其余所有此前未探测的服务，从而最终找到 100% 的归一化服务（尽管速度非常慢）。

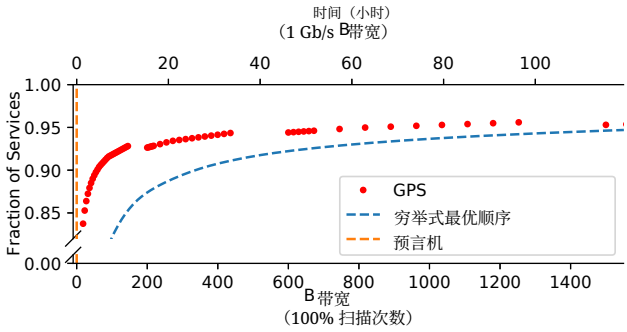
6.4 GPS 与机器学习的比较

如第 2 节所述，Sarabi 等人 [36]使用先进的 XGBoost 分类器 [18]，针对给定端口预测有响应的 IP。不幸的是，由于 (1) 该分类器的运行时间开销过高，以及 (2) 在 99.99% 的全部 65K 端口中，缺少用于训练稳健模型的 1000 万个 IP 地址样本 [36]，因此无法将该分类器部署到全部 65K 端口上。尽管如此，我们仍将 GPS 与 XGBoost 进行基准比较，因为后者是与本研究最相关的已有工作。

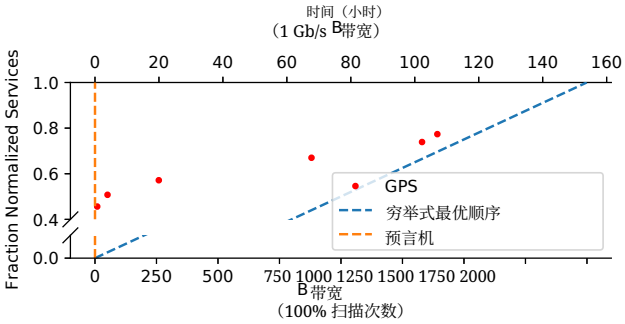
方法。 由于 Sarabi 等人的 XGBoost 扫描器并未开源，我们根据其论文中展示的结果对 GPS 进行基准测试。我们采用 XGBoost 扫描器中表现最佳的实现：一系列按照端口扫描最优顺序排列的模型，并将端口响应作为输入特征。我们将 GPS 配置为使用 Censys 数据集的 0.5% 作为种子大小（相当于 Sarabi 等人用于训练模型的 IP 数量）进行评估；Sarabi 等人也使用了该数据集。我们将 GPS 的步长设为 /16，以平衡覆盖率和准确率。我们针对 Sarabi 等人评估的全部 19 个 TCP 端口和协议进行了评估。由于他们的研究提供了不同覆盖步长下的指标，



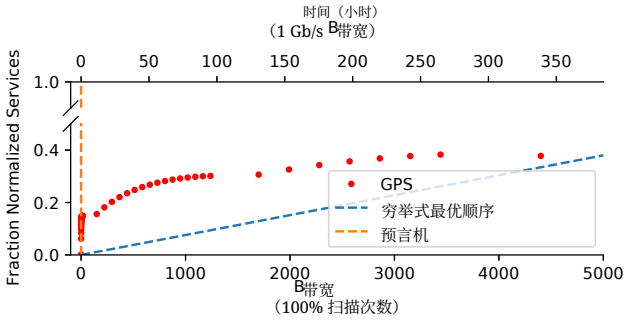
(a) 服务发现（Censys）： 与最优端口顺序探测相比，GPS 使用少 21× 的带宽即可找到 94% 的服务（2K 个端口，100% 扫描，2% 种子）。



(b) 服务发现（LZR）： 与最优端口顺序探测相比，GPS 分别使用少 6× 和少 2× 的带宽，即可找到全部服务的 92.5% 和 95%。（所有端口，1% 扫描，0.5% 种子）。



(c) 归一化服务发现（Censys）： 与最优端口顺序探测相比，GPS 使用少 100× 的带宽，即可在 2K 个端口中找到 46% 的归一化服务；但要找到 67% 的归一化服务时，节省的带宽仅为 50%。（2K 个端口，100% 扫描，2% 种子）。



(d) 归一化服务发现（LZR）： 与最优端口顺序探测相比，GPS 分别使用少 15× 和少 1.7× 的带宽，即可找到 17% 和 38% 的归一化服务。（所有端口，1% 扫描，0.5% 种子）。

图2：发现服务——使 在各种训练集和测试集下，GPS 能够发现高达 85% 的归一化服务和 99.8% of 发现所有服务所需的带宽更少 比最优端口顺序探测。

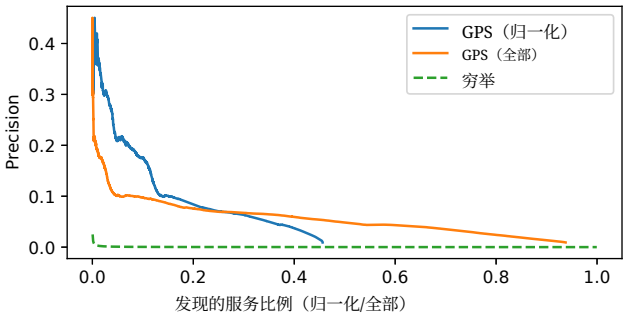


图 3：GPS 精度——GPS 最多可发现 94% 的所有服务和 46% 的归一化服务，并且精确率始终比穷举探测高出一个数量级以上。

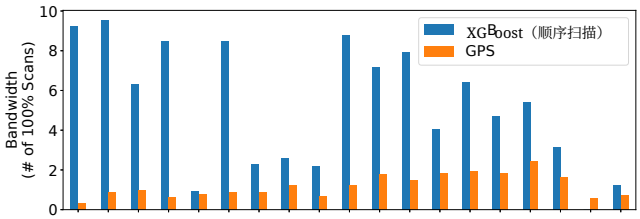
我们在 GPS 能达到的最大覆盖率水平下评估 XGBoost 扫描器[1]（即平均覆盖率为 98.8%）。

[1]使用 GPS，种子大小为 0.5%，步长为 /16，可实现以下最大覆盖率：端口 23、25、119、445、465、587、993、3306、7547、8080、88889 为 99%；端口 21、22、80、443 为 98%；端口 143、995、5432 为 94%；端口 2323 为 94%。

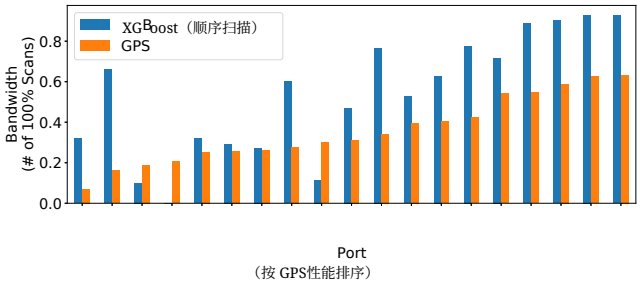
与 GPS 扫描所有主机上的一组最小预测服务以找出剩余服务的方式类似，XGBoost 扫描器也依赖一组先前扫描的 IP 地址和端口来预测其他服务。在图4a中，我们绘制了 XGBoost 扫描器收集所有所需先验信息（遵循其最优扫描顺序）所需的带宽，并将其与 GPS 扫描最小预测服务集所需的带宽进行比较。我们注意到，尽管 Sarabi 等人没有直接透露其模型的预测准确率，但带宽与准确率相关：准确预测服务所需的带宽越少，模型的预测就越准确。

在所有端口上，GPS 收集其最小预测服务集所需的平均带宽是其他方法的 5.7 倍之少；扫描端口 2323 时，最多可减少 28 倍带宽。扫描端口 80 是 GPS 唯一比 XGBoost 扫描器需要更多带宽的情况；XGBoost 扫描器不依赖最小预测服务集来预测端口 80 上的服务（即它只使用网络层特征，立即预测所有服务）。

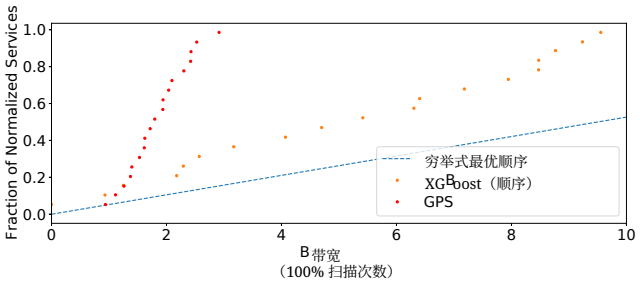
假设已经扫描了预测服务的最小集合（例如，用户原本打算扫描多个端口），我们在图4b中展示了各系统达到目标端口多数覆盖率所需的带宽



（a）扫描最小预测服务集所用的带宽——GPS 节省的带宽最多可达 XGBoost 扫描器的 28 倍。



（b）扫描其余服务所用的带宽——扫描 19 个常用端口中的 16 个时，GPS 比 XGBoost 扫描器节省更多带宽。



（c）归一化服务发现——与 XGBoost 扫描器相比，GPS 使用的带宽减少了 3 倍，即可找到全部归一化服务的 98.5%。

图4：GPS 与 XGBoost 的带宽消耗

。扫描19个评估端口中的16个时，GPS 所需带宽少于 XGBoost。例如，在预测端口2323和5432上的主机时，GPS 所需带宽减少至四分之一。只有扫描端口995、993和465时，XGBoost 所需带宽更少。不过，XGBoost 的性能提升并非没有代价：与 GPS 相比，XGBoost 扫描器需要多消耗3–14倍的先前带宽，才能收集端口995、993和465对应的预测服务最小集合（图4a）。平均而言，GPS 达到相同目标端口覆盖率所需的带宽只有一半。换句话说，在预测服务时，简单的条件概率平均比 XGBoost 机器学习分类器准确率更高。

在计算发现服务所需的全部带宽时，我们在图4c中展示了 GPS 如何以比 XGBoost 扫描器少 3 倍的带宽，找到所有归一化服务中的 98.5%。

6.5 计算复杂度

如果具备适当的计算资源，GPS 的可并行预测算法便能在有限的实际运行时间内发现所有端口上的服务。本节将讨论 GPS 的性能如何取决于种子扫描、带宽和计算资源的可用性。我们在表2中报告了 GPS 的性能明细：将 GPS 配置为使用表 1 中的全部 25 个特征、1% 种子扫描和 /16 扫描步长，以预测所有端口上的服务。

时间。 GPS 将时间花费在三类任务上：扫描、预测和数据传输。GPS 的瓶颈在于带宽。在没有现成种子扫描的情况下，GPS 需要 12.3 天才能完成所有扫描，这受现有扫描工具和可用带宽的限制。收集初始的 1% 种子扫描占所有扫描时间的 97.5%，这是由于随机探测时的精确率较低（即对于大多数端口，精确率低于 0.00001）。然而，如果已经有种子扫描，GPS 可以省去收集初始种子扫描的步骤，从而将总体运行时间缩短 94%。与种子扫描相比，两次预测扫描仅需数小时即可完成，因为它们的精确率比对非常用端口进行穷举探测高出多个数量级。总计，GPS 可预测 280 亿项服务，这些服务需要 8 小时进行扫描。

在单个 CPU 核心上，GPS 完成预测大约需要 9 天 9 小时，比在单个 GPU 上运行的 XGBoost 扫描器快 5.6 倍。然而，将 GPS 的可并行预测算法与高度可并行的计算环境结合后，其墙钟时间可以大幅缩短。将 GPS 的预测算法部署到 BigQuery 后，GPS 可在 13 分钟的墙钟时间内完成所有预测，耗时少 *4* 个数量级 于 XGBoost 扫描器，而成本仅为 75 美分。

遗憾的是，在任何平台上处理大数据通常都会面临数据传输瓶颈。使用无服务器平台需要将数据上传到平台并从平台下载。GPS 将总计 606 GB 的数据上传/下载到 BigQuery 共需 9.1 小时。上传/下载过程中最大的瓶颈是上传/下载带宽；GPS 使用 24 个并行进程时，测得的带宽为 18MB/s–30 MB/s。BigQuery 不对入站数据传输收费，因此 BigQuery 的总成本仅受计算成本限制。Sarabi 等人没有报告将所有必要数据加载到 XGBoost 扫描器 GPU 中所需的时间，而这通常被认为是大规模机器学习计算的最大瓶颈 [6]。

空间。 GPS 所需的内存量与种子扫描的规模、用于预测服务的特征数量，以及计算条件概率所采用的实现方式直接相关。种子扫描本身（包括其中的所有特征）通常相当小；由 LZR [25] 收集的经过筛选的 1% IPv4 65K 端口种子扫描仅为 4 GB。然而，GPS 在预测服务时需要多得多的内存，这取决于条件概率计算的实现方式。例如，第 5.5 节所述的实现方式会将数据与自身进行 JOIN，以找出某个 IP 的特征之间所有成对组合；与

[2]使用附录 B 中的方法。

	带宽	计算时间（单核）	墙钟时间	处理/洗牌的数据	Cost
1% 种子扫描（如有需要）	1.5 Gb/s	–	12天	–	–
种子扫描上传	20 Mb/s	–	3.5 分钟	4 GB	0¢
预测首次服务（PFS）	-	6 天 2 小时	8 分钟（BigQuery）	4 TB	13¢
PFS 下载	1.3 KB/s	–	7 秒	9.3 KB	0¢
PFS 扫描	50 Mb/s	–	20 分钟	–	–
PFS 扫描上传	30 Mb/s	–	34分钟	55 GB	0¢
预测剩余服务（PRS）	–	3天7小时	5分钟（BigQuery）	2.5 TB	62¢
PRS 下载	18 Mb/s	–	8.5小时	547 GB	0¢
PRS扫描	50 Mb/s	–	8小时	–	–
总扫描耗时	–	–	12.3 天	–	–
下载/上传总耗时	–	–	9.1 小时	–	–
总计算耗时	–	–	13 分钟	–	–
总计	–	9 天 9 小时	12.7 天	7 TB	75¢

表2：GPS性能分解—使用 1% 种子扫描和 /16 扫描步长来预测所有端口上的服务时，GPS 的瓶颈在于带宽。使用 BigQuery 时，GPS 会在 13 分钟内完成所有必要的预测。由于 GPS 的扫描精度很高，为了避免数据包拥塞和入站数据包丢弃，两次预测扫描都采用了大幅降低的扫描速率（即 50 Mb/s，而非 1.5Gb/s）。

特征	归一化服务	服务
（端口，端口$_{\text{Protocol}}$）	18.7%	2.0%
Port	14.1%	2.0%
(端口, 端口$_{\text{HTTP}}$ 头部)	9.7%	2.0%
(端口, 端口$_{\text{ASN}}$, 端口$_{\text{HTTP-Body-Hash}}$)	7.7%	2.0%
(端口, 端口$_{\text{HTTP-Body-Hash}}$)	6.1%	2.0%

表3：前5个预测特征—端口的协议是最具预测性的特征，可预测18.7%的归一化服务。

种子扫描及所用特征数量相比，这至少会使内存占用增加 50 倍。GPS 在输出最终的预测服务列表时也需要相当大的磁盘空间；写入预测的 280 亿个 IP和端口对会产生 547 GB 的输出。

部署 GPS 的研究人员若使用高度可并行化的计算环境，且内存比初始种子扫描规模多两个数量级，将获得最大的性能提升。

6.6 哪些特征最具预测性?

通过使用条件概率，可以轻松了解哪些特征最能预测互联网服务。当使用 1% 的种子集运行 GPS，以预测 Censys 数据集所有端口上的服务时，GPS 会选出 402K 个独特的特征 值，认为它们最能预测某项服务。与其他所有协议相比，使用HTTP协议时发现的信息（例如 HTTP标头、HTTP正文哈希）最具预测性，占最具预测性的特征值的 45%。我们在表3中展示了 GPS 识别出的最具预测性的前 5 个特征候选项。在 18.7% 的归一化服务中，主机端口上运行的协议（例如 SSH）（即 (Port, PortProtocol)）最能预测另一个端口是否响应。

GPS 识别出 64 个最能预测服务的独特特征 值元组，其中包括应用层和网络层特征之间的交互，例如：(ASN, TLS 证书)（占归一化服务的 4.4%）、(ASN, SSH密钥)（1.2%）、VNC名称（0.4%）以及 (ASN, FTP横幅)（0.24%）。例如，在 Distributel Network（自治系统编号 1181）中，响应端口 23 且显示 Telnet横幅"Telnet service is disabled or Your telnet session has expired due to inactivity..."的主机中，有 95% 的主机在端口 8082 上承载 HTTP内容；而在 Bizland（自治系统编号 29873）中，对端口 143 的请求以要求 TLS 的互联网邮件访问协议横幅响应的主机中，有 98% 的主机还在端口 2222 上运行 SSH。这些发现表明，GPS 对特征交互进行建模的能力（第 5.2 节），以及不排除任何特征候选项的决策，有助于预测各种网络和端口上的服务。

7 局限性

虽然 GPS 大幅降低了扫描所有互联网服务的门槛，但仍存在一些挑战。

模式挖掘。 GPS 的能力受限于其配置使用的特征（附录 C）以及由此挖掘出的模式。例如，如果检测某种模式需要收集来自多个端口响应并集的特征，GPS 可能无法检测到该模式，因为计算两个以上端口之间的相关性在计算上代价高昂。尽管如此，随着计算并行性日益增强，GPS 的算法采用模块化设计，易于扩展以加入额外的特征相关性计算。

IPv6。 GPS 依赖穷举扫描来获取一组响应式 IP 地址，用于服务预测；由于 IPv6 的搜索空间更大，这种方法并不适用。不过，对于已知的、至少在一个端口上有响应的 IPv6 地址，GPS 可以用来预测这些已知 IPv6 地址上的其他响应服务。

随机主机配置。虽然服务通常呈现可预测的模式，但随机配置的主机始终会带来

预测服务的局限性。例如，在我们的 LZR 扫描中发现的最常见物联网设备（占 21%）FRITZ!Box，其手册指出："出于安全原因，启用通过 HTTPS 访问互联网后，FRITZ!Box 会为 HTTPS 设置一个随机 TCP 端口" [2, 5]。此外，路由器可以轻松地通过随机端口转发服务 [7, 10]。我们在 LZR 扫描中发现，至少 55% 的服务很可能正在通过端口转发（即所有托管服务返回的 TTL 值各不相同），且这一情况出现在 99% 的非常用端口上。

为了解随机主机配置如何影响 GPS，我们设计了一项实验：(1) 使用 95% 的种子集来预测 Censys 数据集中剩余 5% 的服务，由此假设几乎所有模式都已事先获知；(2) 将某个 IP 上的所有服务都视为在该 IP 上至少发现一项服务时即已发现，由此假设特征相关性 100% 可用且 100% 准确；(3) 指定最大的扫描步长（/0），以最大化已发现归一化服务的总比例。在这些理想条件下，使用少于穷举扫描所需的带宽即可发现所有归一化服务中的 80%；这一比例略低于 GPS，这是因为随机测试集规模较小。这些结果表明，在"真实世界"环境中，GPS 的表现接近最佳可实现水平，同时也揭示了任何智能互联网范围扫描系统未来将面临哪些根本局限性。

8 结论

在这项工作中，我们介绍了 GPS，这是一种智能扫描系统，能够在无需先验知识的情况下，以可扩展且高效的方式预测所有 IP 和端口上的服务。我们展示了一个看似简单、基于条件概率的预测框架，如何以比领先的机器学习实现快几个数量级的速度，取得更高的准确率。GPS 使用 131× 比穷举扫描更少的带宽，就能找到所有端口上 92.5% 的服务，同时精确度还高出 204× 。我们将 GPS 作为开源工具发布，希望研究界从此能够以远低于穷举扫描的成本，发现此前遗漏的数十亿项服务。

致谢

作者感谢 Tatyana Izhikevich、Katherine Izhikevich、Kimberly Ruth、Deepak Kumar、Pratiksha Thaker、Deepti Raghavan、斯坦福大学安全与网络研究组的成员、我们的审稿协调人 Lixia Zhang，以及匿名评审，感谢他们提供富有见地的讨论和意见。本研究部分由美国国家科学基金会（资助编号 CNS-1823192）、Google., Inc.、NSF 研究生奖学金 DGE-1656518 和斯坦福研究生奖学金资助。

参考文献

[1] BigQuery：云数据仓库。https://cloud.google.com/bigquery。访问日期：2021年-08-28。[2] 无法通过家庭网络的 HTTPS 访问用户界面。https://en.avm.de/service/knowledge-base/dok/FRITZ-Box-4020/3439_无法通过家庭网络访问用户界面。访问日期：2022-01-26。[3] LZR 数据集：识别意外的互联网服务。https://scans.io/study/lzr。访问日期：2022-01-26。[4] Free。https://www.free.fr/freebox/。访问日期：2021年-09-09。[5] FRITZ!Box 7490 帮助。https://service.avm.de/help/en/FRITZ-Box-Fon-WLAN- 7490/015/hilfe_internet_remote_https。访问日期：2021-08-13。[6] GPU 速度很快！数据集才是瓶颈。https://towardsdatascience.com/gpus-are-fast-datasets-are-your-bottleneck-e5ac9bf2ad27。访问日期：2021-09- 06。[7] 如何设置端口转发——多种路由器品牌通用指南。https://www.noip.com/support/knowledgebase/general-port-forwarding-guide/。访问日期：2021-09-09。[8] 通过过滤互联网伪服务，超越噪声。https://censys.io/blog/beyond-noise-by-filtering-pseudo-services/。访问日期：2021-08-19。[9] 推荐系统。https://github.com/microsoft/recommenders。访问日期：2021-09-06。[10] 了解端口转发。https://stevessmarthomeguide.com/understanding-port-forwarding/。访问日期：2022-01-26。[11] 通用互联网 BigQuery 数据集。https://support.censys.io/hc/en-us/articles/360056063151-Universal-Internet-BigQuery-Dataset。访问日期：2021-09-09。[12] ZGrab 2.0 - GitHub。https://github.com/zmap/zgrab2。访问日期：2019-12-14。[13] M. Antonakakis、T. April、M. Bailey、M. Bernhard、E. Bursztein、J. Cochran、Z. Durumeric、J. A. Halderman、L. Invernizzi、M. Kallitsis 等。了解 Mirai 僵尸网络。载于 USENIX 安全研讨会，2017年。[14] S. Bano、P. Richter、M. Javed、S. Sundaresan、Z. Durumeric、S. J. Murdoch、R. Mortier 和 V. Paxson。扫描互联网以检测存活状态。ACM SIGCOMM 计算机通信评论，2018年。[15] B. Beurdouche、K. Bhargavan、A. Delignat-Lavaud、C. Fournet、M. Kohlweiss、A. Pironti、P.-Y. Strub 和 J. K. Zinzindohoue。一团混乱的现状：驯服 TLS 的复合状态机。载于 IEEE 安全与隐私研讨会，2015年。[16] R. Beverly。在互联网上运行 Yarrp：随机化高速主动拓扑发现。载于 ACM互联网测量大会，2016年。[17] S. Checkoway、R. Niederhagen、A. Everspaugh、M. Green、T. Lange、T. Ristenpart、D. J. Bernstein、J. Maskiewicz、H. Shacham 和 M. Fredrikson。双椭圆曲线在 TLS 实现中的实际可利用性。载于 USENIX 安全研讨会，2014年。[18] T. Chen 和 C. Guestrin。Xgboost：一种可扩展的树提升系统。载于 ACM SIGKDD 国际知识发现与数据挖掘会议，2016年。[19] Z. Durumeric、D. Adrian、A. Mirian、M. Bailey 和 J. A. Halderman。基于全互联网扫描的搜索引擎。载于 ACM 计算机与通信安全会议，2015年。[20] Z. Durumeric、F. Li、J. Kasten、J. Amann、J. Beekman、M. Payer、N. Weaver、D. Adrian、V. Paxson、M. Bailey 等。Heartbleed 问题。载于 ACM互联网测量大会，2014年。[21] Z. Durumeric、E. Wustrow 和 J. A. Halderman。ZMap：快速全互联网扫描及其安全应用。载于 USENIX 安全研讨会，2013年。[22] P. Foremski、D. Plonka 和 A. Berger。Entropy/IP：揭示 IPv6 地址中的结构。载于 ACM互联网测量大会，2016年。[23] O. Gasser、Q. Scheitle、P. Foremski、Q. Lone、M. Korczyński、S. D. Strowes、L. Hendriks 和 G. Carle。广袤网络中的集群：理解并消除 IPv6 地址列表的偏差。载于 ACM互联网测量大会，2018年。[24] C. A. Gomez-Uribe 和 N. Hunt。Netflix 推荐系统：算法、商业价值与创新。ACM 管理信息系统汇刊（TMIS），2015年。[25] L. Izhikevich、R. Teixeira 和 Z. Durumeric。LZR：识别意外的互联网服务。载于 USENIX 安全研讨会，2021年。[26] M. Kula。用于用户和物品冷启动推荐的元数据嵌入。arXiv 预印本 arXiv:1507.08439，2015年。[27] D. Kumar、K. Shen、B. Case、D. Garg、G. Alperovich、D. Kuznetsov、R. Gupta 和 Z. Durumeric。综合考量：家庭网络中物联网设备的分析。载于 USENIX 安全研讨会，2019年。[28] F. Li、Z. Durumeric、J. Czyz、M. Karami、M. Bailey、D. McCoy、S. Savage 和 V. Paxson。您遇到了漏洞：探索有效的漏洞通知。载于 USENIX 安全研讨会，2016年。[29] J. Ma、K. Levchenko、C. Kreibich、S. Savage 和 G. M. Voelker。意想不到的协议推断方法。载于 ACM互联网测量大会，2006年。[30] Y. Ma、B. Narayanaswamy、H. Lin 和 H. Ding。实时的时间上下文推荐。载于 ACM SIGKDD 国际知识发现与数据挖掘会议，2020年。[31] B. Marczak、J. Scott-Railton、K. Berdan、B. Razzak 和 R. Deibert。揭秘 Candiru：另一家雇佣间谍软件供应商进入公众视野。技术报告，2021年。[32] B. Marczak、J. Scott-Railton、S. Prakash Rao、S. Anstis 和 R. Deibert。兜兜转转：揭露网络间谍公司 Circles 的客户。技术报告，2020年。[33] J. McInerney、B. Lacker、S. Hansen、K. Higley、H. Bouchard、A. Gruson 和 R. Mehrotra。探索、利用与解释：利用多臂老虎机实现个性化、可解释的推荐。载于 ACM 推荐系统会议，2018年。[34] G. C. Moura、C. Ganán、Q. Lone、P. Poursaied、H. Asghari 和 M. van Eeten。ISP 地址空间的动态性有多大？迈向全互联网 DHCP 地址变更估计。载于 IFIP 网络会议，2015年。

[35] A. Murdock、F. Li、P. Bramsen、Z. Durumeric 和 V. Paxson。全互联网 IPv6 扫描的目标生成。载于*ACM互联网测量大会*，2017年。[36] A. Sarabi、K. Jin 和 M. Liu。智能互联网探测：使用自适应机器学习进行扫描。2021年。[37] SHODAN。互联网连接设备的搜索引擎。https:// www。shodan。io/.[38] K. Vermeulen、J. P. Rohrer、R. Beverly、O. Fourmaux 和 T. Friedman。Diamond-miner：全面发现互联网拓扑中的菱形结构。载于*USENIX网络系统设计与实现研讨会 (NSDI)*，2020年。

附录
附录是未经同行评审的补充材料。

智能扫描推荐系统

专有推荐系统已成功地为 Netflix [24]、Spotify [33]，和 Amazon [30] 向数百万用户推荐了数百万件商品。这些模型的成功启发我们使用推荐系统，为 IP 地址推荐/预测有响应的端口。我们探索了 30 多种不同的开源推荐模型 [9]，发现现有推荐系统虽然允许为用户（即 IP）和/或商品（即端口）分配特征，却无法直接为用户与商品之间的交互（即 IP、端口元组）分配特征。因此，无法轻松表示每种服务的应用层特征。此外，大多数模型不支持添加新特征，而这些特征可能是在成功预测后发现的；若要将其加入模型，就不得不承担重新训练模型带来的额外计算开销。

为了评估开源模型能否成功预测服务，我们扩展了一种广受欢迎的开源混合推荐器 LightFM [26]，以根据 IP地址预测可响应的端口。我们在 GitHub[3]上提供了该模型的实现，包括其全部参数。由于该框架无法为特定服务（即（IP，端口）对）分配特征，我们只能为 IP 地址和端口编码特征。我们尝试为每个 IP 地址分配不同的网络层特征（例如自治系统、/16 子网、/20 子网），并为每个端口分配一个二元特征（用于标示端口号是否由 IANA 分配）。我们使用 LZR 数据集在全部 65K 端口上训练模型（使用 0.8% 的 IPv4 种子集），并让模型为测试集中的每个 IP 地址生成 100 个端口预测（即类似于生成 100 次"100% 预测扫描"）。该模型最多只能发现全部服务的 47%——其表现始终不如按优先发现最多服务的顺序对端口进行穷举探测——以及 1.5% 的归一化服务。由于推荐系统施加的特征限制，以及该模型表现不佳，我们决定不再继续研究调整推荐系统以预测所有端口上的服务。

B 筛选真实服务
相关工作表明，大量 IP 地址托管着"伪服务" [8]。伪服务通常是已成功加载的 HTTP 或 HTTPS 网页，但网页本身会显示一条消息，称该网页上不存在任何服务。我们进行了一次

[3]https://github.com/斯坦福-esrg/recommender-system-gps

网络特征	% 服务 最具预测性
ASN	36%
/16	20%
/18	8%
/19	8%
/17	8%
/20	7%
/21	6%
/22	4%
/23	3%

表4：网络特征—将 GPS 配置为使用 IP地址的 /16–/23 子网时，对于大多数服务，IP地址的自治系统编号和 /16 最具预测性。

2021年3月，我们对 IPv4 地址空间中随机抽取的 1% 子集的全部 65K 端口进行了 LZR [25] 扫描，发现对于全部 65K 端口中的 96%，绝大多数服务都属于在超过 1,000 个连续端口上提供伪服务的主机。

为确保 GPS 不会学会预测"伪服务"，GPS 的种子集必须过滤掉所有伪服务。超过 80% 的伪服务只需通过两步即可过滤：首先，移除数据中预期会动态变化的字段（例如 HTTP 日期字段、HTTP Cookie 字段、TLS 随机字节）；其次，移除主机上所有与过滤后数据相同的服务。然而，长尾伪服务更难通过指纹识别并过滤。例如，数据中包含意料之外的随机字符串（例如"事件 ID"、时间戳），会导致数据和内容长度略有不同。现有的过滤方案往往复杂且不完善 [29]。因此，我们会过滤掉提供超过 10 项服务的任何主机——该方法能够以 100% 的召回率和 99% 的精确率识别伪服务。本文中的所有结果均假设已过滤伪服务。

C 网络层特征候选项
GPS 的最终实现配置为仅使用对大多数服务最具预测性的网络层特征。减少网络层特征的数量可以降低 GPS 预测算法的计算复杂度。为了确定哪些网络层特征最能预测服务存在，我们最初将 GPS 配置为使用 /23–/16 之间的所有子网大小以及自治系统编号。GPS 使用 LZR 数据集中的 0.5% 种子集构建概率模型，我们据此分析哪些网络特征对所有服务最具预测性。表4 显示，IP 的自治系统编号和 /16 子网最能预测服务存在。较大的子网更能预测服务存在，这并不令人意外：较大的子网更有可能由种子集中的多个主机共享。

D 参数调优
D.1 改变步长
GPS 通过调节扫描步长，在预测期间最大限度地减少带宽。随着步长减小，发现

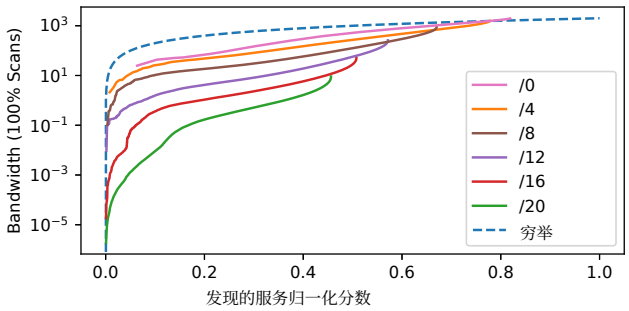
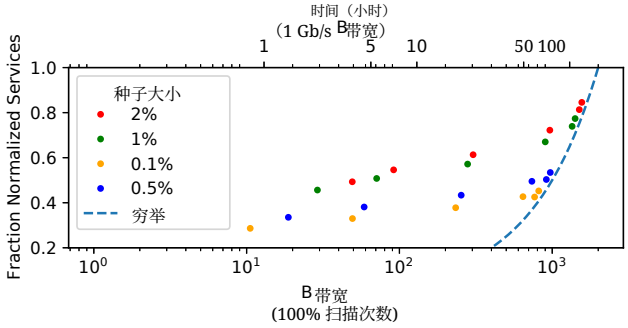


图 5：改变步长（Censys）——较小的扫描步长在最初查找服务时可以节省更多带宽，但最终找到的服务少于较大的扫描步长。

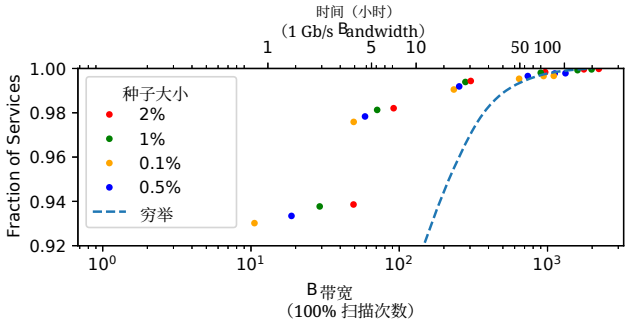
服务的精确率会提高，从而降低所需的总体带宽。例如，如图 5 所示，使用 /12 的扫描步长找到归一化服务中最先出现的 25%，所需带宽比使用 /20 的扫描步长多一个数量级。没有任何 GPS 配置能在带宽用量优于穷举探测的情况下找到超过 82% 的归一化服务。

D.2 改变种子大小

种子大小决定了 GPS 已经见过多少种独特的服务模式。我们在图 6 中绘出了所需带宽（包括收集种子所需的带宽）以及发现的服务比例。当优化目标是发现归一化服务时，图 6a 显示，若带宽预算超过 30 个 100% IPv4 扫描，2% 的 IPv4 种子大小总能比更小的种子大小发现更高比例的归一化服务。这表明，在更大规模的种子扫描中发现的模式，对于发现归一化服务至关重要。然而，当优化目标是发现最大比例的服务时，较小的种子大小就足够了（图2a），这表明只需较小的种子大小，就能检测到最能预测热门服务的模式。



（a）归一化服务发现



（b）服务发现

图 6：改变种子大小（Censys）——较大的种子大小会提高 GPS 找到的归一化服务所占的比例，但不会显著影响 GPS 找到的全部服务所占的比例。