课程论文题目

6Forest: An Ensemble Learning-based Approach to Target Generation for Internet-wide IPv6 Scanning

摘要

IPv6 目标生成是对 Internet 范围的调查进行快速 IPv6 扫描的关键步骤。然而由于离群地址和短视的分裂指标导致的空间分区不当，现有技术普遍存在命中率低的问题。为了解决这个问题，文章提出了 6Forest，这是一种基于集成学习的 IPv6 目标生成方法，它从全局角度出发，对异常地址具有弹性。给定一组已知地址，6Forest 首先将其视为初始地址区域，然后使用最大覆盖拆分指示器将 IPv6 地址空间迭代划分为更小的区域。在一轮空间划分之前，它为每个区域建立一个森林结构，并利用增强的隔离森林算法去除离群地址。最后，它从划分的地址区域中预扫描样本，并根据结果生成 IPv6 地址。在八个大型候选数据集上的实验表明，与 IPv6 全球扫描中的最先进方法相比，6Forest 可以在低预算扫描方面实现高达 116.5% 的改进，在高预算扫描方面实现高达 15 倍的改进。

关键词：全网扫描;IPv6; 异常值检测; 集成学习

1 引言

自 2011 年世界 IPv6 日以来，由于移动互联网和云计算的强烈需求，IPv6 近年来得到广泛实施和采用。2021 年 7 月，超过 30% 的谷歌用户通过 IPv6 访问他们的服务，互联网上的 IPv6 路由条目数量正在迅速增加。对如此庞大的地址空间进行探测和分析的全互联网调查并非易事，这阻碍了对基于 IPv6 的互联网进行有效的网络资产评估和风险分析。传统的异步扫描工具，如 ZMap 和 Masscan，涉及拓扑发现、IP 地址分析、和地理定位等先进技术，以促进高性能网络调查。然而这些工具是为 IPv4 网络设计的，在处理 IPv6 地址空间时通常会面临效率问题。事实上，仅基于这些蛮力方法，可能需要数千万年才能对整个 IPv6 地址空间进行全面扫描。

为了消除这种缺陷，IPv6 目标生成被用作高效 IPv6 扫描的必要步骤。通过用一组已知地址（即种子）对 IPv6 地址空间的结构进行表征和建模，可以生成具有较高活跃概率的候选地址，从而缩小探测范围并显着加快扫描速度。

2 相关工作

使用种子生成 IPv6 目标的研究早在 2012 年就开始了。Barnes 等人假设已知的活动地址提供有关寻址方案使用的信息。种子信息有助于发现更多新地址的这一假设已成为后续研究的基础，并被最近的实验结果所强化。迄今为止，相关研究对种子中的语义信息和结构信息均进行了开发。

在 6GC-VAE、6VecLM 和 6GAN 等基于语义信息的方法中，首先根据 IPv6 向量空间映射技术（即 IPv62Vec）将种子转换为向量。在 IPv62Vec 之后，矢量数据集将用于训练深度神经网络（例如，Transformer、Variational Autoencoder 和 Generative Adversarial Network）。此外，IPv6 地址中的每个半字节及其所在位置都将被视为一个词，IPv6 地址将被构造为一个句子。IPv6 目标生成问题转换为已解决的文本生成问题。然而深度神经网络的巨大计算成本意味着这些方法无法扩展到大规模扫描。

对于第二类，种子的结构信息主要用于确定扫描区域或指导目标生成。Foremski[1]等人介绍了Entropy/IP，这是一种从种子中学习模式的算法，它利用经验熵将 IPv6 地址的相邻半字节分组为段，并使用贝叶斯网络对不同段值之间的统计依赖性进行建模。这种学习到的统计模型用于生成扫描的目标地址。Murdock[2]等人提出 6Gen，它假设具有高密度种子的地址空间更可能具有未被发现的活动地址。6Gen 扩展每个种子作为每个簇的中心，通过保持最大种子密度和最小规模来生成目标地址。Liu[3]等人提出 6Tree，它利用由给定种子结构形成的空间树来划分 IPv6 地址空间。6Tree 根据已知活跃地址的个数计算空间树上节点的密度，然后它根据给定节点的密度生成目标地址。Hou[4]等人提出 6Hit 并首先将强化学习应用于 IPv6 主动扫描。6Hit 根据各区域扫描奖励动态分配预算。通过反馈，6Hit 将后续搜索方向优化到高密度区域。

考虑到基于语义信息的模型难以应对大规模 IPv6 扫描及其可解释性挑战，文章参考种子的结构信息来生成目标。第二类技术的主要弱点是最左分裂指标和“害群之马”（即离群种子）导致的大规模扫描命中率相对较低。为此，这项工作致力于应对这些挑战并推进命中率的极限。[5]。

3 本文方法

3.1 本文方法概述

图 1说明了 6Forest 的主要工作流程。它首先用一组已知的活跃 IPv6 地址（即种子）对整个 IPv6 地址空间进行空间划分，根据最大覆盖分裂指标将种子迭代划分为更小的簇。空间分区后，6Forest 利用隔离森林算法检测离群种子并将其从地址区域中移除。最后，6Forest 从最终地址区域中抽取一些目标（< 给定预算的 1%），并对其进行预扫描以估计地址区域的命中率。那些具有高估计命中率的区域将被优先用于目标生成。

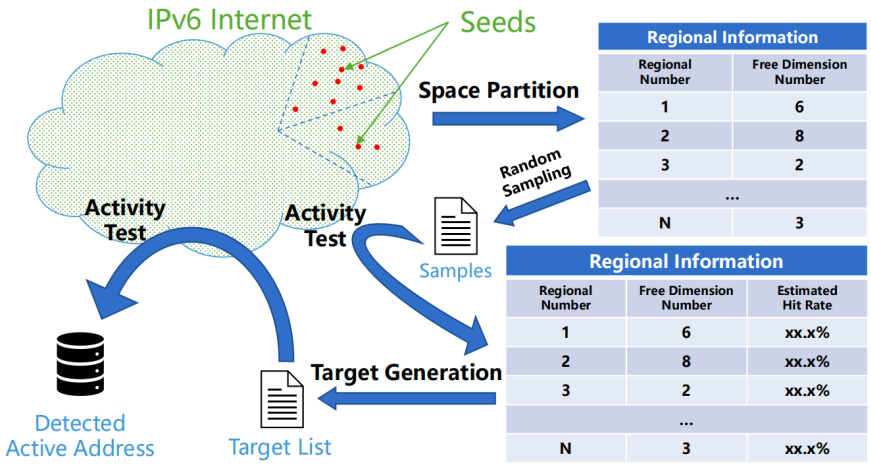

图 1: 6Forest 工作流程图

3.2 相关定义

定义 1：种子（地址）区域。经过空间划分后，所有给定种子 S 的集合被划分为小规模的种子簇，除了那些只有一个种子的簇外，即是所谓的种子（地址）区域。孤立的种子在所有维度上都有固定的半字节，不能指导目标生成。

定义 2：自由维度和固定维度。我们认为整个 IPv6 地址空间有 32 个维度，每个维度的范围是从 0x0 到 0xf。对于给定的地址区域，所有种子都具有相同半字节值的那些维度是所谓的固定维度，否则它们是自由维度。按照惯例我们使用通配符“*”来表示不确定的半字节（自由维度）。

定义 3：离群种子。在给定的地址区域中，那些具有唯一地址模式的种子将被视为异常值。

定义 4：覆盖。给定地址区域的一个自由维度的非唯一半字节的频率之和。直观地说，覆盖的目的是为了找到最小化孤立种子数量的分裂指标。假设一个区域包括 K 个种子，其第 i 个维度的半字节向量为 V_i，半字节 j 的频率定义为 $|Vi == j|$:

$$K = \sum_{j=0}^{15} |V_i == j|$$

经过归一化后，第 i 个维度的覆盖可以表示如下：

$$\text{Cover}_{\ i} = \frac{\sum_{j=0,|V_i==j|>1}^{15} |V_i == j|}{K}$$

如图 2 所示，6Forest 为每个自由维度分配一个覆盖值，并采用第一个最大覆盖自由维度（第 17 个）而不是最左边的自由维度（第 15 个）作为后续种子划分的分裂指标。

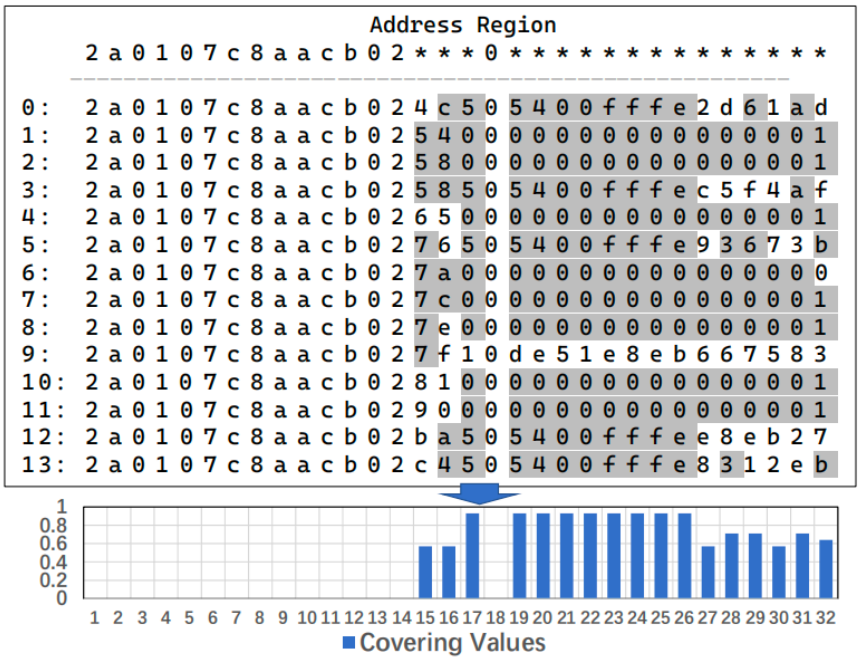

图 2: 地址区域的覆盖示例

3.3 相关定义

3.4 空间划分

目前最先进的方法中的空间划分是通过递归地将种子划分为更小的种子簇来构建树结构“空间树”来实现的。除了离群种子的挑战外，不合适的分裂指标（即粗糙的最左边的自由维度）也是整个空间划分的缺点。

为此，本文提出了一个新的指标来评估所有自由维度的优先级作为分裂指标，即覆盖。覆盖表示给定自由维度上所有种子的非唯一半字节数。为了充分利用种子，最大覆盖突出了一个自由维度，其中种子分裂后孤立的种子最少。此外，理想种子区域中的自由维度并不严格落后于固定维度，简单的最左分裂指示器可能导致种子的错误划分。另外，现有的方法采用深度优先搜索策略，递归拆分当前叶子节点进行“空间树”的扩展，即保留整个空间树结构，很难并行执行。为此，本文利用广度优先搜索策略和先进先出结构（即队列）而不是树结构来进行空间划分。在空间划分时，队列中只保存那些可用于后续划分的地址区域，从理论上降低了空间复杂度。此外，6Forest 中的队列可以被多个 CPU 共享（即资源池化），可以显着提高空间划分的效率。

在一轮种子划分中，6Fores 首先从队列中取出节点，并分别所有自由维度分配覆盖值。6Forest 利用具有最大覆盖值的自由维度的半字节作为簇代表，将节点的当前种子划分为对应的子节点，即具有相同半字节值的簇，然后将它们压入队列中。此外，如果有几个自由维度共同分配了最大覆盖值，6Forest 会选择最左边的一个。

值得强调的是，对于大地址区域，6Forest 的空间划分与 6Tree 或 6Hit 类似，因为分裂指标通常是当前区域最左边的自由维度，可以快速呈现初始种子的粗略分类。对于小地址区域，6Forest 选择最小化孤立种子数量的分裂指示符，而不管其位置如何。总之，6Forest 的空间划分是站在整个种子的角度进行的。

3.5 离群种子检测

我们已经定义了离群种子，并指出它导致了扫描空间的膨胀。然而，如何自动检测这些异常仍未解决。在本节中，6Forest 利用隔离森林算法过滤掉异常值，并为 IPv6 目标生成提供没有异常值的地址区域。基本假设是异常值最先被隔离，并且它们的半字节值在大多数自由维度上都是唯一的。6Forest 首先在每个自由维度上构建一个深度受限的隔离树，然后将隔离森林的权重分别累加为种子的异常分数，最后那些异常分数高于阈值的种子将被视为异常值。

图 3 是离群种子检测算法的一个示例, 使用的地址区域是图 2 中的地址。6Forest 首先构建相应的隔离森林。在图 2 中，三个典型的隔离树分别表示种子在第 15、17 和 28 维上的划分结果。并且将对限制深度树的孤立种子进行评分以表示异常值。请注意，异常值（即第 9 号）具有唯一的地址模式，并且可以在大多数树中被隔离，但不是全部（例如，第 15 维的隔离树）。

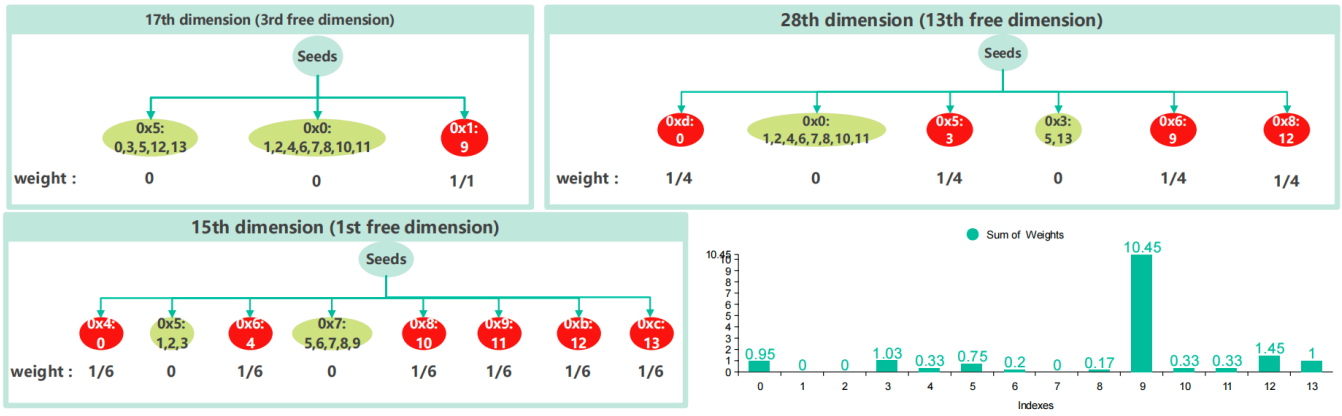

图 3: 异常值检测的隔离森林示例

一个直截了当的想法是利用整个森林的综合结果，即集成学习。然而简单地累积异常值表示的分数是不明智的，因为正常地址区域在一些自由维度上也有不同的半字节，隔离树通常会将它们的种子误认为是异常值。换句话说，分裂指标本身可能是不正确的，如果大多数种子都被隔离，其权重应该降低。

为了便于计算，假设在隔离树中有 k 个孤立的种子时，权重被动态设置为 $\frac{1}{k}$ 。考虑具有 f 个自由维度和 m 个种子的地址区域，所有自由维度上的异常值表示的分数之和等于 f，并且种子的期望分数（阈值）是 $\frac{f}{m}$。因此，那些大于种子区域期望分数的种子是异常值，需要被移除。对于图 2 中具有 13 个种子和 17 个自由维度的给定区域，6Forest 认为那些权重超过阈值的种子将被视为异常值（例如，权重为 10.45 的 9 号种子）。

4 复现细节

4.1 与已有开源代码对比

文章作者公开了关于空间划分和离群种子检测的源代码，复现的代码主要参考了现有代码，并对有关目标地址生成的代码进行补充。

4.2 实验环境搭建

- 种子地址：经过筛选后的 2022 年开源的 IPv6 数据集 10 万。
- 扫描主机：1 台具有 8G 内存、Intel i7-6700 处理器的 Ubuntu22.04 虚拟机。
- 复现代码：Python 实现的 6Forest 代码。
- 探测存活技术：ICMPv6, 使用工具 ZMAPv6[6]。

4.3 创新点

原先代码在空间划分阶段只判断当前地址区域内种子数量是否小于设定阈值，没有很好地判断种子区域是否划分完毕，例如图 4所示。编号 4-6 号种子（IEEE-derived）和 9-12 号种子（Low-byte）显然属于不同的地址模式，但算法却将其划分在了一个地址区域内，这将会导致后续的扫描地址区域膨胀。

```
********Region**********
2a04**********000000000000000***
2a04274711d20000000000000000001   0
2a0427471c82000000000000000001    1
2a042fc7000000000000000000000001  2
2a0445407109670000000000000000001 3
2a045200fff70000000000000000016c  4
2a045200fff70000000000000000017   5
2a045200fff700000000000000002ee   6
2a045200fff700000000000000004f0   7
2a045200fff700000000000000af7     8
2a04c00700000000000000000000001   9
2a04c00700000000000000000000010   10
2a04c00700000000000000000000015   11
2a04c00700000000000000000000005   12
```

图 4: 现有源代码结果

```
********Region**********
2a042**7****0000000000000000001
2a04274711d20000000000000000001   0
2a0427471c82000000000000000001    1
2a042fc7000000000000000000000001  2
********Region**********
2a0445407109670000000000000000001
2a0445407109670000000000000000001 0
********Region**********
2a04c00700000000000000000000**
2a04c00700000000000000000000001   0
2a04c00700000000000000000000010   1
2a04c00700000000000000000000015   2
2a04c00700000000000000000000005   3
********Region**********
2a045200fff70000000000000000***
2a045200fff70000000000000016c     0
2a045200fff700000000000000017     1
2a045200fff700000000000002ee      2
2a045200fff700000000000004f0      3
2a045200fff70000000000000af7      4
```

图 5: 改进后结果

改进后的代码在空间划分阶段添加了对于自由维度判断，如果自由维度大于某一设定值，则需继续划分，直至地址区域内种子数量和自由维度小于对应阈值。

5 实验结果分析

实验从开源的种子地址随机抽取作为种子地址集合, 通过这种方式实验构造了数量为 100000 的种子地址作为地址生成算法的输入，使用改进前后的 2 种算法分别生成了 20000 和 200000 个新地址，

实验重复 3 次取平均值，使用 ZMAPV6 探测存活情况得到图 6所示的存活率。从结果横向比较分析，两者整体命中率都比较接近，与论文在部分种子集上的结果相似，但达不到在其他大部分种子集上的命中率。尽管我们的实验和论文的工作都采用了相似的数据集，但是纵向比较仍然只能作为参考。首先，随着时间的推移，收集的 IPv6 地址集合本身就会有大量地址不再存活，并且在不同的时间和地点探测的存活率也是不一样的。其次，研究者是如何处理和筛选数据的方法也不尽相同，没办法完全比较各自输入的种子地址集合差异。综上，虽然很难有完全精确地纵向比较这些基于种子的地址探测技术的实验，但是通过在相同的实验条件下的横向对比，可以发现不同探测技术的优缺点，尤其是在时间性能和命中率方面的好坏。

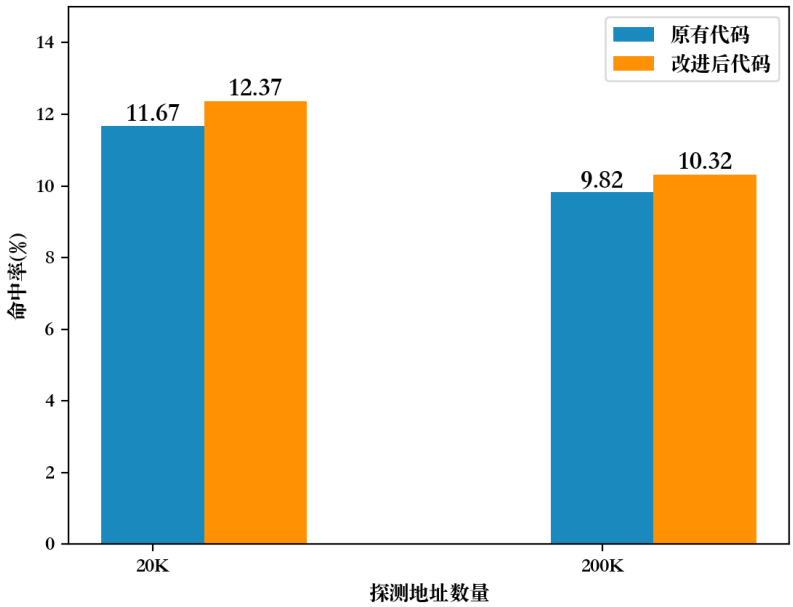

图 6: 实验结果示意

6 总结与展望

6Forest 在分裂指标的选择上采用了一种新颖的最大覆盖度量和增强的隔离森林算法来自动过滤异常种子，生成细粒度的地址区域，有效地减小了扫描空间的大小。在实现过程中未能详尽地测试高预算下复现结果的真实情况，只测试了单种子集，结果准确性没有很好，而且代码改进前后的效果并不明显。未来考虑采用带有额外信息的 IPv6 情报作为输入，利用这些额外的信息来优化地址生成算法也许能提高探测命中率。

参考文献

[1] FOREMSKI P, PLONKA D, BERGER A. Entropy/ip: Uncovering structure in ipv6 addresses[C]//Proceedings of the 2016 Internet Measurement Conference. 2016: 167-181.

[2] MURDOCK A, LI F, BRAMSEN P, et al. Target generation for internet-wide IPv6 scanning[C]//Proceedings of the 2017 Internet Measurement Conference. 2017: 242-253.

[3] LIU Z, XIONG Y, LIU X, et al. 6Tree: Efficient dynamic discovery of active addresses in the IPv6 address space[J]. Computer Networks, 2019, 155: 31-46.

[4] HOU B, CAI Z, WU K, et al. 6hit: A reinforcement learning-based approach to target generation for internet-wide ipv6 scanning[C]//IEEE INFOCOM 2021-IEEE Conference on Computer Communications. 2021: 1-10.

[5] YANG T, CAI Z, HOU B, et al. 6Forest: an ensemble learning-based approach to target generation for internet-wide IPv6 scanning[C]//IEEE INFOCOM 2022-IEEE Conference on Computer Communications. 2022: 1679-1688.

[6] GASSER O. Evaluating network security using Internet-wide measurements[D]. Technische Universität München, 2019.