SN 计算机科学（2022） 3:126
https://doi.org/10.1007/s42979-022-01017-z

原始研究

使用机器学习识别和计算NAT背后的主机

Sanjeev Shukla[1] - Himanshu Gupta[1]

收到。2021年9月29日 / 已接受。1 January 2022 / 在线发表: 11 January 2022

摘要
NAT是一个部署大量主机（通常在局域网内）使用单个或一组有限的公共IP地址的过程。NAT提供的匿名性是合法用户的特权，但它是恶意用户/攻击者隐藏而不被发现的便利场所，这有助于他们执行拒绝服务（DoS）攻击。因此，识别隐藏在NAT后面传播恶意行为的主机对于取证调查是必要的。在本文中，我们提出了一种基于机器学习（ML）的算法来识别和计算NAT背后的主机总数。这是通过在网络流量中通过流量级别的统计识别模式和检测主机的匿名性来实现的。拟议工作的新颖之处在于计算主机，并将结果与传统的基于签名的方法进行比较。获得的结果显示了更高的准确性（使用三个数据集平均93%），大大超过了基于签名的方法结果（约75%）。此外，该模型还通过向其提供恶意网络流量数据集进行测试，以识别和计算流量是来自于单个还是多个主机。获得的较高准确率（约92.2%）反映了该模型能够预测和计算正确的恶意主机数量，且误报率最低。在这项工作中，我们通过在统计网流记录上应用基于ML的技术来识别和计算NAT背后的主机数量。所提出的方法的优点是，在捕获头信息时不会出现与隐私有关的问题，而且它不依赖于操作系统、IP地址或端口号。

关键词 NAT - DoS - ML - 网络安全

简介

网络地址转换（NAT）是互联网服务提供商（ISP）部署的一个非常普遍的设备，以提供互联网服务。NAT[1]是用一个公共IP地址替换多个主机的私有IP地址的过程。这样做是为了在NAT网关后面的小型私人局域网（LAN）中部署大量的系统，通过NAT网关使用互联网。

本文是由Rajiv Misra, R K Shyamsunder, Alexiei Dingli, Natalie Denk, Omer Rana, Alexander Pfeiffer, Ashok Patel和Nishtha Kesswani客串编辑的 "通信网络的网络安全和隐私 "专题集的一部分。

✉Sanjeev Shukla sanjufcc@iitr.ac.in

Himanshu Gupta
himanshugpt3@gmail.com

[1]印度理工学院
计算机科学系，罗尔基，乌塔拉坎德邦247667，印度

IP接口。网关后面的机器使用私人IP地址进行内网活动，并使用网关的公共地址进行互联网。因此，NAT是任何网络工作设置中的一个重要元素，并通过为私人局域网中的用户提供匿名性，为IP地址的短缺和隐私问题提供解决方案而获得了巨大的欢迎。安装在私人和公共网络的入口和出口点，它还通过隐藏内部网络拓扑结构、网络监控、用户匿名和内容过滤来协助解决一些安全问题。

激励

一家安全服务提供商公司发表的白皮书在其研究报告"NAT作为安全提供商的神话"中说明，大量的攻击和威胁来自使用NAT的ISP网络。这在移动网络中也有发现[2]。同样，最近发表在ACM上的研究文章在2021年9月讨论了NAT在DOS攻击中的脆弱性[3]。因此，这项工作的动机是为了开发

一个ML模型，可以识别NAT设备后面的主机，并帮助计算它们。

在NAT中面临的挑战

因此，NAT有助于解决安全和隐私问题方面的一些挑战，但它也有自己固有的安全缺陷。最大的问题是识别隐藏在NAT设备的公共接口后面的单个主机[4]。对于恶意的使用，它提供了完全的匿名性。NAT提供的这种匿名性是合法用户的特权，但它也是网络攻击或恶意用户的安全港。

宿主识别的需要

从网络外部识别或追踪优先局域网中的确切主机（这是网络攻击或恶意软件传播的源头）的挑战是困难的。每一次试图从广域网（WAN）找到攻击源，都会把他们引向网络网关（NAT）的公共接口，这就成为检测源的瓶颈。在没有网络管理员的帮助下，要进一步检测和找到确切的主机是不可能的。另外，现在，ISP在向用户提供互联网连接时，根据活动连接的数量提供服务（除了其他基于带宽的计划）。在基于活跃用户的连接数的情况下（例如，最多10个活跃使用），ISP需要知道NAT后面的活跃用户数，以确保他们不超过付费用户连接配额。在这种情况下，对主机的计数是必不可少的。其他应用程序，如流媒体服务器，也需要主机计数以更好地服务其用户，因为服务器提供有限的用户连接，由于服务器过载。在NAT用户的情况下，一个用户请求（隐藏大量的匿名用户）可能会用实时流协议（RTSP）请求压倒服务器，拒绝其他用户的服务器访问。这可能会模仿DOS攻击，而用户数对于此类服务是至关重要的[5, 6]。

论文投稿

到目前为止，传统的基于ML的方法是从网络流量中检测路由器NAT或NAT设备。到目前为止，还没有做主机的计数。我们的论文提出了一种基于ML的算法来识别和计算NAT后面的主机总数。这是通过在网络流量中寻找模式，通过流量级别的统计和检测主机的匿名性来实现的。拟议工作的新颖性在于计算NAT后面的主机总数，并将其结果与最先进的基于签名的方法进行比较。我们在本文中的主要贡献是。

1. 使用提议的基于ML的方法识别和计算隐藏在NAT后面的主机总数。
2. 为模型提供恶意主机数据集，以计算恶意主机的数量。这将有助于在检测到恶意流量的情况下，但我们不知道它是来自单个主机还是多个主机。
3. 与其他最先进的技术进行比较分析，以确定该模型的功效。

因此，目前工作的目标是开发一个ML模型，它可以很容易地检测到NAT设备（在私人局域网）后面的主机，并帮助计算主机的总数。这种模型的意义在于，根据网络流量的模式，利用流量级别的统计数据，准确预测主机的匿名性。

纸张组织

本文的结构如下。相关工作
"部分讨论了过去在NAT环境下的主机检测和识别领域所做的工作，并与我们提出的工作相比较。建议的方法
"一节详细介绍了建议的工作、工作流程、解释和使用的数据集。应用的方法
"部分显示了应用的方法，分类器的选择，性能指标等。结果和讨论
"部分讨论了所获得的结果，最后，"结论和未来工作
"部分包含了结论并提供了一些未来的工作方向。

相关作品

主机识别技术--主机检测--
主要可以通过两种方法实现，主动探测或被动探测。主动探测技术向目标系统发送一组查询并分析其响应。它需要与目标主机进行双向通信，而且在所有的网络配置中可能不被允许。此外，它是侵入性的，可检测的，并消耗网络资源。另一方面，被动探测技术捕获网络流量，以便日后分析。它不产生自己的流量，而是被动地监测和观察网络流量。因此，它是非侵入性的，不可察觉的，并且可以追溯分析捕获的网络流量。因此，我们将只关注被动技术。被动技术又被分为基于签名的方法和基于ML的方法。

基于签名的技术

基于签名的技术用于检测NAT后面的主机，就是要找到一组参数（签名），在没有IP地址的情况下，可以唯一地定义主机。该

整个过程是被动地捕获网络流量，并进行标题分析以识别主机。在过去，研究人员提出了许多方法来检测NAT背后的主机，如Cohen[7]提出了一个模型，使用不同协议字段之间的相关性来识别源机器，Maier等人[8]使用IP
TTL和HTTP用户代理字符串来估计连接到NAT设备的主机数量，Mongkolluksamee等人[9]提出了一种被动方法来计算具有类似IP地址的主机，即NATed设备，Wicherski等人[10]提出了一种被动方法，用于计算具有相似IP地址的主机。
Paul等人[11]提出了一个轻量级的实时灵活框架NATFILTERED，该框架使用时间戳（TCP）来识别主机，而不依赖于IP地址；Paul等人[11]提出了一种基于TCP SYN的方法，用于识别虚拟机（VM）环境中使用的未经授权的操作系统的恶意活动；Park等人[12]提出了一种技术，使用TCP/IP协议的头域来检测NAT背后的全部主机及其操作系统类型。

基于ML的技术

表1详细显示了关于基于ML的方法检测NAT和它背后的主机的文献调查。Abt等人[13]。建议使用从网流中提取的行为统计学来检测胭脂红的NAT设备。二元分类器使用九个不同的特征将流量分类为NAT或非NAT。Verde等人[14]提出了一个指纹框架，通过抽取主机样本并对其进行剖析以训练HMM分类器来识别主机。编码后的网流进入训练集，被用户检测模块用来汇总时间间隔，并将用户分类为连接或不连接。这种方法的局限性在于，首先必须对用户进行训练，这降低了它的适用性，而且在使用TOR或VPN隧道时，其性能也会下降。

Gokcen等人[15]提出了基于ML的方法，从网络工作流量中检测NAT设备背后的主机的恶意行为。它部署了C4.5算法，这是一个决策树分类器，还使用了Naive Bayes统计分类器，该分类器应用贝叶斯定理来获得一个给定类别的条件概率。它进一步使用WEKA，一个开源工具，并通过混淆矩阵参数来衡量性能。它在没有应用（有效载荷）信息和加密环境下也能很好地工作。Komark等人[16]提议被动地利用HTTP访问日志中的IP特征来检测NAT行为。他们从HTTP日志中提取了所有独特的特征，如

表1 过去使用ML进行主机和NAT检测的工作

参考文献	目标	应用的分类器	NAT检测	主机剖析	主机计数	数据集类型	使用的特征数量	准确度（平均%）
Abt等人[13]。	要检测一个使用行为统计的胭脂红NAT设备	SVM和DT	是	没有	没有	净流量	9	89.95
Varde等人。[14]	指纹识别NAT后面的用户	HMM	没有	是	没有	净流量	-	90
Gokcen等人[15]。	通过分析交通流来识别NAT行为	NB和C 4.5	是	没有	没有	净流量	40	DR=98
Komarket al.[16]	NAT检测使用统计学行为分析	证券公司	是	没有	没有	HTTP访问原木	8	95
Khatouni等人。[17]	NAT检测和主机识别	邓小平	是	没有	没有	净流量	10	F1得分=96
Lee等人[18]。	NAT设备方法	邓小平	是	没有	没有	净流量	-	F1得分=90
建议的	使用端口响应模式进行识别计算NAT后面的主机总数							

税率	没有	没有	是	净流量	1411 得分 = 92

用户代理、操作系统及其版本、联系的IP地址、持续连接数、浏览器家族及其版本、发送的HTTP请求数、上传和下载的字节数等，用于分析和NAT识别。

Khatouni等人[17]提出了一个基于ML的模型，用于使用Netflow统计数据被动地测量NAT行为。他们在八个不同的Netflow特征上比较了十种机器学习算法，四个特征来自文献[15]，四个来自流行的Netflow软件，得出结论：Tranalyzer特征集及其性能比其他软件好得多。他们的结果是，使用Tranalyzer的决策树（DT）分类器提供的解决方案以最小的计算成本获得了最高的F1分数。Lee等人[18]提出使用端口响应模式来识别NAT设备。这是通过在模型上应用超视距学习实现的，该模型利用了NAT和非NAT设备的端口响应的不对称模式。这种方法的局限性是在收集端口响应时需要大量的时间，主动探测的挑战，如探测的阻断通过防火墙或IDS将它们视为威胁。

建议的方法

在这项工作中，我们的目的是识别隐藏在NAT设备后面的主机数量并对其进行统计。然后我们对传统的基于签名的方法和基于ML的方法进行比较分析。我们想把我们的工作与类似的基于ML的方法进行比较，但大多数论文（如表1所示）都是检测NAT而不计算主机。对于提议的方法，我们在与八个多类分类器比较其性能后，使用了决策树（DT）分类器。对于基于签名的方法，我们根据文献[9, 10, 12]编写了python代码，作为最先进的技术。对于这两种方法，我们都部署了相同的数据集。

工作流程流程

代表基本工作流程的框图如图1所示，"建议的方法"部分涉及的步骤如下。

1. **第一步是捕捉**样本网络（或测试设置）的网络流量。这是用wireshark完成的，因此得到的文件是pcap格式的。
2. pcap文件被送入tranalyzer应用程序以提取统计特征。Tranalyzer默认给出了105个特征。
3. 接下来，我们应用预处理和特征选择方法来提取数据集中最重要的特征。

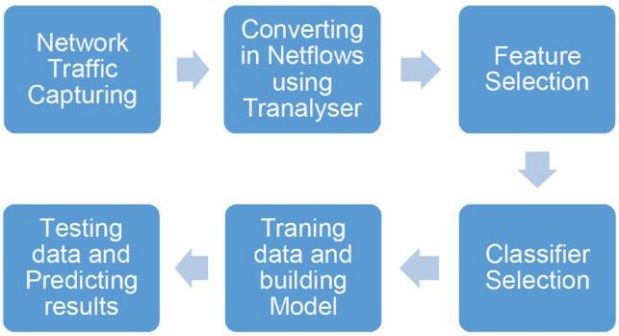

图1 拟议的工作流程框图

4. 对不同的多类分类器进行了测试，以找到最适合我们问题陈述的分类器。
5. 在选择分类器后，数据集被训练以建立测试模型。
6. 最后，应用测试数据集来获得模型的结果。

数据集

在拟议的工作中，我们已经采取了三种不同的数据集。两个来自网上，一个来自我们的实验室设置。第一个数据集是在线提供的，属于ICS实验室[19]。它包含来自ICS实验室的1116个不同主机的非NAT-ted流量。我们将其命名为ICS数据集。该数据集包含一个大小约为400MB的pcap文件。Trana-lyser被用来将pcap文件转换为csv文件，并提取了关键的明显特征。这个数据集在预处理后包含460万个流量，这是一个大数据集；因此，我们选择它来训练我们的模型。

第二个在线资源是恶意软件流量分析（MTA）网站上的一个pcap文件[20]。这是一个小型数据集（称为MTA数据集），其中包含143个恶意主机的NAT-ted和非NAT-ted流量。测试这个数据集的目的是看所提出的方法是否能够检测和计算恶意主机。

第三组数据是在我们的实验室里用73台不同的主机产生的，时间为6小时。我们的设置如图2所示，包括一个交换机和73台PC，每台PC有8GB内存和一个i7六核处理器（时钟速度4.3GHz）。这73台电脑用于生成网络流量，并在NAT设备前后放置捕获设备，以便对NAT化和非NAT化的流量进行分析。流量是通过一个脚本产生的，该脚本在网络浏览器中打开多个URL。使用wire-shark以pcap格式收集数据，并使用tranalyzer提取关键参数。提取的参数经过预处理并转换为csv文件。我们把这个数据集命名为LAB数据集。

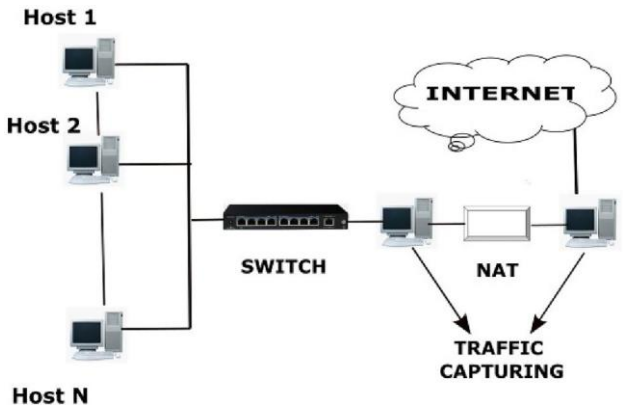

图2 我们实验室的设置

使用的软件

- Tranalyzer。它是一个流量生成器工具，支持制作大型数据包转储。它提供了从pcap文件生成统计数据的功能。市场上有许多软件，包括Net- mate, Argus, Tstat等。我们在工作中使用Tranalyzer，因为根据文献[17]，与其他软件相比，它的结果更好。
- Wireshark。它是免费以及开源的数据包分析器，可用于流量捕捉和分析。我们使用wireshark作为工具，为我们的LAB数据集捕获pcap格式的数据包。
- Weka。Weka是一个开源工具，它包含一个可视化工具和数据分析和预测建模的算法集合。它还支持聚类、分类和回归等多任务。

特征选择

由此获得的数据集被转换为网络流。这是用Tranalyzer完成的，它将pcap文件转换为dis-files文件。
基于数据流的统计特征。使用基于流量的方法的好处是，我们不再需要IP地址和端口号。因此，任何类型的偏见
所提出的方法消除了NAT实施带来的影响。

拟议的基于ML的方法中使用的特征

这是在主机识别过程中的一个重要步骤。在pcap格式的数据集上应用Tranalyzer，可以得到105个统计特征。

预处理

预处理包括数据准备、过滤、清洗、特征提取和选择过程。没有经过仔细检查的数据集会产生误导性的结值。同时，为了保持每个特征的重要性，将字符串格式的特征转换成相应的整数值。因此，这导致了47个特征的子集。

特征选择

特征选择是任何数据模型的最关键部分。在这里，我们的目标是在47个特征中选择最优化的特征，这些特征应该是独立于用户行为和网络环境的，以产生一个高效的数据模型，该模型应该足够强大，可以在不同的环境下工作。为此，我们使用了两种方法的组合：（1）过滤法和（2）包裹法。第一种方法是过滤法，这是在包装法之前需要的，以使包装法的计算效率高。在过滤方法中，我们尝试了Chi-Square（CS）和Information gain（IG）技术，我们发现CS的结果并不适合我们的问题陈述，因为它依赖于显著性水平。因此，我们选择了IG，它在结果上更适合我们的问题。另外，根据文献，大多数作者都表明IG是一种很好的技术，可以参照我们的问题陈述来选择最佳属性[17]。此外，我们还做了一个简单的比较，以了解基尼指数（GI）和基于熵量子的IG之间的差异。GI有利于较大的分布，并且容易实现，而IG则有利于较小的分布，具有多个特定值的小计数。GI主要用于CART算法，而IG则用于ID3、C4.5和J48算法。另外，GI对分类目标变量进行"成功"或"失败"的操作，只进行二元分割；而IG则计算分割前后的熵差，并指出元素类别中的不纯性。因此，IG被应用于计算归一化的平均杂质，使用公式(1)

$$\text{增益}(S, A) = \text{熵}(S) - \sum_{v \in \text{值}(A)} \frac{Sv}{} \text{熵}\, Sv, S \quad (1)$$

其中Values(*A*)是属性A的所有可能值，Sv是属性A有值v的S的子集，熵用公式（2）计算

$$\text{熵}\ (S) = - \left(P \oplus \log 2\ P \oplus + P \ominus \log 2\ P \ominus \right) \quad (2)$$

。
果。因此，我们对数据集进行了预处理，去除多余的特征和恒定的特征。

其中N是S的比例，P是S中负面例子的比例。

图3显示了基于信息增益值的所有特征的排名，其中*X轴*显示了特征编号（47个特征中的每个都用数字表示），*Y轴*显示了信息增益值。正如我们从图3中观察到的，增益值在22个特征之后急剧下降。因此，我们选择了对我们模型影响最大的前22个特征。

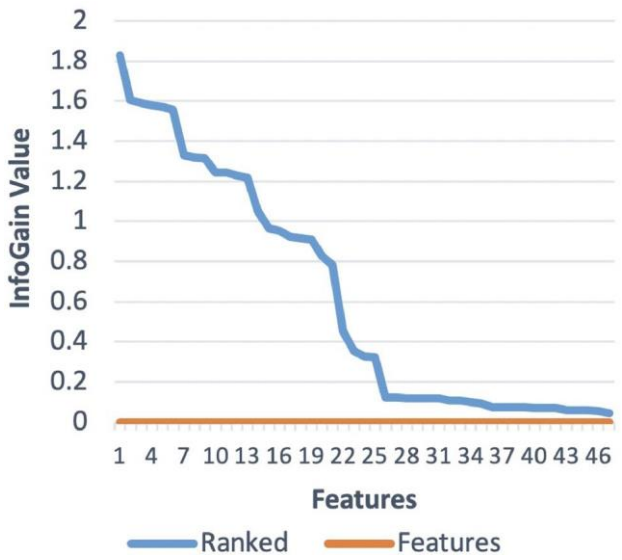

图3 特征排名

在过滤方法之后，我们转向了包装方法。在包装方法中，我们主要采用遗传算法和单点交叉法，目的是为我们的问题找到最优化的特征集。图4显示了代表包装方法过程的方框图，该方法涉及的步骤如下。

1. **第一步，我**们通过过滤法得到初始特征集。
2. 在第二步，使用交叉技术从过滤方法后留下的所有可用属性中选择一部分特征。
3. 在第三步，在通过交叉选择的特征子集上实施ML算法。
4. 在第四步，分析使用所选特征的算法的性能，并从第二步开始重复这一过程，直到我们得到最佳的特征集。

我们发现，如表2所示，wrapper方法产生了一个由14个特征组成的子集，这对我们的问题陈述来说是最好的结果。因此，我们为给定的问题选择了这14个特征，因为与47个特征相比，它提高了模型的整体性能。

表2 所选功能

特点	描述
dstPortDestination	port tcprttAckTripMaxTCP
	ACK trip maximum
tcpRTTAckTripAveTCP	ACK trip average
tcpAveWinSzTCP	average effective window size
tcpMaxWinSzTCP	最大有效窗口大小 tcpInitWinSzTCP
	初始有效窗口大小 tcpMinWinSzTCP
	最小有效窗口大小 tcpRTTAckJitAveTCP
	ACK往返平均抖动 tcpAckFaultCntTCP
	ACK数量故障计数
tcpWinSzThRtTCP	计数比率低于WINMIN阈值 flowIndFlow
	索引
dstPortClassNPort	基于目的端口的分类 pktAsmPacket
	流不对称性
tcpWSTCP	窗口比例因子

应用的方法

对于建议的方法

在这一点上，我们使用ICS数据集来训练我们的模型，其中包含来自1116台主机的流量。我们通过源IP地址来识别每个主机，并将它们标记为一个独特的类别。一旦为每个流量分配了类别，我们就在ICS数据集的帮助下训练我们的模型。在训练完我们的数据模型后，我们使用了不同的数据集，其中包含来自多个主机的NAT-
ted流量来测试我们的训练模型。然后，我们训练的模型在所有选定的特征的帮助下，将每个流量分类到一个类别。然后对独特的类别进行统计，以计算主机的数量。开始时，我们首先选择一个适当的分类器。

分类器的选择

为数据的分类和预测而选择的ML分类器是基于其性能。由于存在多个类别，所以选择多类分类器进行预测。接下来，在ICS数据集上对8个多类ML分类器进行了比较，其中有14个特征集（从特征提取中获得），采用了k-
fold交叉验证（k=10）。表3列出了用于比较的分类器。

图4 包裹方法流程图

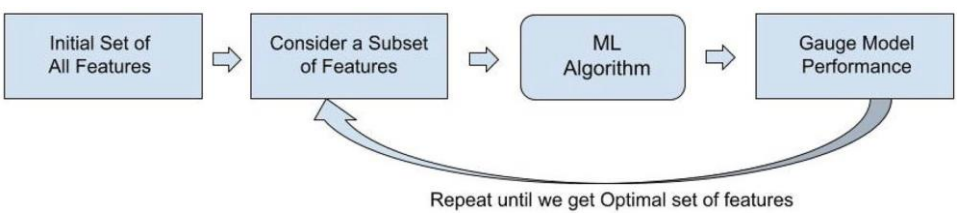

表3 多级分类器列表

S. no.	算法简述	
1	奈何贝叶斯	NB
2	霍夫丁树	HT
3	随机树	访谈
4	决策树	决策树
5	随机森林	射频
6	K最近的邻居	KNN
7	多层感知器	MLP
8	支持向量机	SVM

如图5所示，所有选择的分类器的性能分别用精度、召回率和F1分数值进行评估。因此，比较结果显示决策树（DT）和随机森林（RF）分类器胜过其余分类器。两者的结果很相似，但模型的建立时间（训练时间）却大不相同。DT建立模型的时间为18秒。RF的模型建立时间为
182秒，性能的提高仅为0.1%（与DT性能相比）。该系统的配置
用来测试我们的ML模型的机器是。操作系统-Windows 10, 处理器-Intel(R) Core(TM) i5-7200U CPU @ 2.7 GHz，系统-64位操作系统和处理器，内存-16GB和HDD-1
TB。因此，选择了DT，因为与RF相比，模型建立的时间非常少，而且得到的结果是
是更快的，其性能几乎与射频相似。

训练和测试模式

一旦分类器被选中，我们就使用ICS数据集来训练模型。首先，我们应用十倍交叉验证。该
用MTA和LAB数据集进一步测试该模型，以检查其性能。

应用的性能指标

使用四个指标来评估性能，这些指标是基于混淆矩阵计算的。真阳性（TP）是模型正确预测到其主机的净流量的结果。假阳性（FP）是指模型完全正确地预测了其他主机到其主机的净流量。假阴性(FN)是指模型错误地预测了流向其他主机的净流量。真阴性（TN）被认为是0，因为不存在被拒绝的网流。关于性能评估的指标的细节如下。

- **精度。它被定**义为TP与TP和FP的总和之比。

$$\text{精度} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{3}$$

- 召回率。它被定义为一个分类器检测所有可用的阳性样本的能力。它被表述为

$$\text{召回} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{4}$$

- F1得分：它被定义为召回率和精确率的HM（谐波平均值），由以下公式得出

$$f1\ 分 = \frac{2 * \text{精度} * \text{召回率}}{\text{精度} + \text{召回率}}. \tag{5}$$

图5
不同机器学习分类器的比较

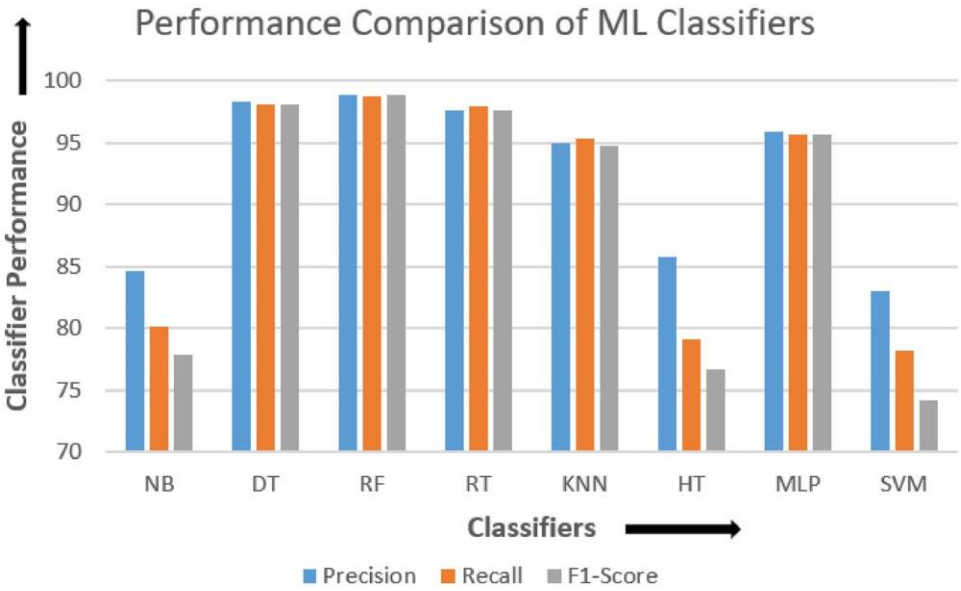

表4 拟议方法的结果

数据集	精度(%)	召回率(%)	F1得分(%)	准确率(%)
MTA数据集	92.1	92.6	92.1	92.2
LAB数据集	87.0	89.0	88.0	89.0

- **准确度**：**它是真阳性和真阴性的**总和与数据包总数的比率。

$$\text{准确度} = \frac{TP + TN}{tp + tn + fp + fn}. \quad (6)$$

结果和讨论

在这一节中，我们介绍了所提出的方法的结果，并与基于签名的方法进行了比较。

拟议的ML方法的性能评估

如表4所示，使用pre-cision、recall、F1 score和准确率来评估所提出的方法的性能。由此得到的结果表明，所使用的数据集表现较好，平均准确率为91%。

当ICS数据集被用于训练和测试时，我们得到了98%的平均准确率，这表明当训练和测试用同一数据集完成时，它的表现要好很多。另一方面，LAB数据集的性能略有下降，如表4所示。这反映出，由于训练数据集（ICS数据集）与测试数据集（LAB数据集）不同。因此，如果训练和测试数据集不同，模型的性能就会降低。此外，有恶意流量的MTA数据集被正确识别，以预测正确的主机数量。这意味着，如果恶意流量被送入我们的算法，它将确定恶意流量是来自单个主机还是多个主机（通过计算主机的总数）。

拟议方法与基于签名的方法的比较分析

通过我们的代码实现的基于签名的方法（使用相同的数据集）的性能评估结果为75%（最高），与文献结果相似（大约）。在图6中，使用两种方法的准确性进行的比较性能评估清楚地表明，基于ML的拟议方法（蓝色）优于基于签名的方法。

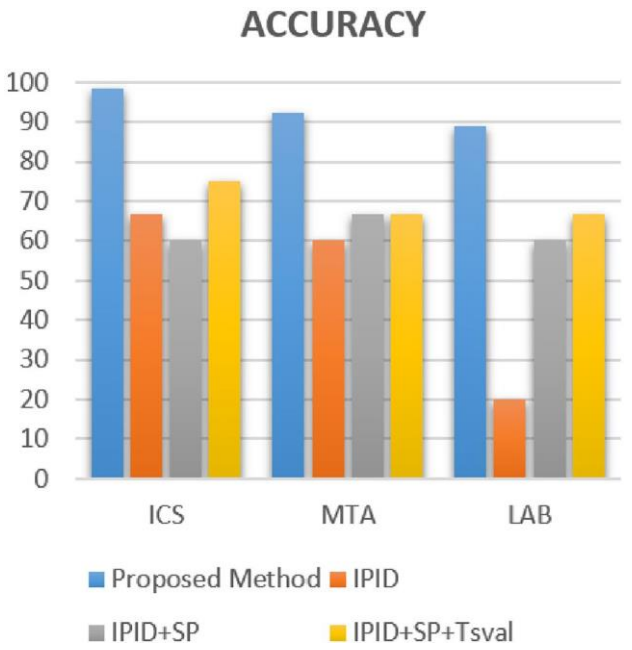

图6 签名和建议方法的比较

基准和拟议框架中的比较点

如表5所示，比较结果显示，大多数基准研究获得了0%或33.33%的分数，而我们提出的框架工作获得了66.66%的分数。以前的基准研究主要集中在NAT设备的检测上，而我们提出的框架则集中在NAT设备后面的主机上。此外，我们提出的框架足够强大，可以处理以前的基准研究中缺乏的恶意流量。

结论和未来工作

在本文中，我们描述了一种基于ML的技术，它与操作系统无关，通过分析统计网络流量来计算NAT后面的主机数量。三个不同的数据集被用来进行测试。通过Tranalyzer处理数据集来获得统计特征。应用预处理和特征选择方法，获得了14个最重要的特征。对不同的多类分类器进行了比较（基于它们的性能），发现Decision Tress是它们中最好的。这样得到的结果显示出更好的结果，所有使用的数据集的平均性能为91%。该模型在计数恶意主机时也显示出良好的结果，当输入恶意流量时。这将有助于确定恶意流量是否来自单一主机。

表5 基准检查表

检查表问题	Abt等人[13]	Varde等人[14]	Y.Gokeen等人[15]。	Komark等人[16]	Khatouni等人[17]	Lee等人[18]	拟议工作
处理NAT设备检测		x					x
处理主机数	x	x	x	x	x	x	
处理恶意流量	x	x	x	x	x	x	
分数	33.33%	0%	33.33%	33.33%	33.33%	33.33%	66.66%
差异	33.33%	66.66%	33.33%	33.33%	33.33%	33.33%	-

或多个主机。此外，与基于签名的技术相比，其结果更胜一筹。当前技术的局限性是，如果使用n个类来训练模型，它将预测*m*个类，其中*m*≤*n*。这个局限性可以通过增加*n*来克服（通过训练网络中的最大主机）。

未来的工作将包括检测恶意流量，并从网络流量中计算隐藏在NAT后面的良性和恶意主机的总数。目前的工作也可以通过上传云端进行在线训练，并在动态网络环境下更新模型。

资金 不适用。

申报

利益冲突 代表所有作者，通讯作者声明没有利益冲突。

参考文献

1. Ishikawa Y, Yamai N, Okayama K, Nakamura M. An identification method of individuals behind nat router with proxy authentication on http communication.In: 2011 IEEE/IPSJ international symposium on applications and the internet, IEEE; 2011. pp.445-450.
2. 网络地址转换为安全的神话。2016年。https://www.f5.com/services/resources/white-papers/the-myth-of- network-address-translation-as-security。2016年2月10日访问。
3. Akashi S, Tong Y.动态网络地址转换对拒绝服务攻击的脆弱性。In:4th International con- ference on data science and information technology, ACM; 2021. pp.226-230.
4. Tang F, Kawamoto Y, Kato N, Yano K, Suzuki Y.基于探针延迟的自适应端口扫描，用于NAT后面具有私有IP地址的Tot设备。J IEEE Netw. 2020; 34(2):195-201.
5. Tekeoglu A, Altiparmak N, Tosun A. Approximating the number of active nodes behind a nat device.In: 2011 Proceedings of 20th international conference on computer comm networks (ICCCN), IEEE; 2011. pp.
6. Meidan Y, Sachidananda V, Peng H, Sagron R, Elovici Y, Shabtai A.检测家庭NAT后面连接的脆弱物联网设备的新方法。J Comput Secur.2020;97: 101968.
7. Cohen MI.网络地址翻译前的来源归属。J Digit Investig.2009;5:138-45.
8. Maier G, Schneider F, Feldmann A. 住宅宽带网络中的Nat使用。In:国际被动和主动网络测量会议，Springer；2011。32-41.
9. Mongkolluksamee S, Fukuda K, Pong P. 通过观察TCP/IP现场行为来计算Natted主机。In: 2012 IEEE interna- tional conference on communications (ICC), IEEE; 2012. pp.1265-1270.
10. Wicherski G, Weingarten F, Meyer U. IP无关的实时流量过滤和使用TCP时间戳的主机识别。In:第38届IEEE本地计算机网络会议，IEEE；2013年，第647-654页。
11. Tyagi R, Paul T, Manoj B, Thanudas B. 用于检测企业中未授权操作系统的包检查。IEEE Secur Priv.2015;13(4):60-5.
12. Park H, Shin RBS, Lee C. 利用IP和TCP的多个字段识别NAT设备背后的主机。In: 2016 Interna- tional conference on information and communication technology convergence (ICTC), IEEE; 2016. pp.484- 486.
13. Abt S, Dietz C, Baier H, Petrovic S. 使用来自netflow的行为统计数据进行被动的远程源NAT检测。In:IFIP自主基础设施、管理和安全国际会议，Springer；2013年，第148-159页。
14. Verde N, Ateniese G, Gabrielli E, Mancini L, Spognardi A. No nat'd user left behind: fingerprinting users behind NAT from net-flow records alone.In: 2014 IEEE 34th international conference on distributed computing systems, IEEE; 2014. pp.
15. Gokcen Y, Foroushani VA, Heywood A. Can we identify NAT behavior by analyzing traffic flows?In: 2014 IEEE security and privacy workshops, IEEE; 2014. pp.
16. Komarek T, Grill M, Pevny T. 使用http访问日志的被动NAT检测。In: 2016 IEEE international workshop on information forensics and security (WIFS), IEEE; 2016. pp.
17. Khatouni A, Zhang L, Aziz K, Zincir I, Heywood N. 使用机器学习探索NAT检测和主机识别。在：2019年第15届网络和服务管理国际会议（CNSM）上，IEEE；2019年，第1-8页。
18. Lee S, Kim SJ, Lee J, Roh B. 基于监督学习的利用端口响应模式的快速、隐蔽和主动NAT设备识别。发表于MDPI的Symmetry杂志。2020;12(9):1444. https://doi.org/10.3390/sym12091444.
19. 工业网络安全会议的数据集，ICS LAB。(2015)。https://4sics.se/。2020年2月1日访问。
20. 恶意软件分析网站的数据集。2017年。https://malwa re-traffic-analysis.net/。Accessed 9 Feb 2020.

出版商提示 《斯普林格-

自然。 施普林格对出版的地图和机构隶属关系中的管辖权要求保持中立。