



计算机工程
Computer Engineering
ISSN 1000-3428, CN 31-1289/TP

《计算机工程》网络首发论文

题目: 基于侧信道特征的安全代理流量分类
作者: 高平, 广晖, 陈熹, 李光松
DOI: 10.19678/j.issn.1000-3428.0058954
网络首发日期: 2020-08-25
引用格式: 高平, 广晖, 陈熹, 李光松. 基于侧信道特征的安全代理流量分类. 计算机工程. <https://doi.org/10.19678/j.issn.1000-3428.0058954>



网络首发: 在编辑部工作流程中, 稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定, 且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式 (包括网络呈现版式) 排版后的稿件, 可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定; 学术研究成果具有创新性、科学性和先进性, 符合编辑部对刊文的录用要求, 不存在学术不端行为及其他侵权行为; 稿件内容应基本符合国家有关书刊编辑、出版的技术标准, 正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性, 录用定稿一经发布, 不得修改论文题目、作者、机构名称和学术内容, 只可基于编辑规范进行少量文字的修改。

出版确认: 纸质期刊编辑部通过与《中国学术期刊 (光盘版)》电子杂志社有限公司签约, 在《中国学术期刊 (网络版)》出版传播平台上创办与纸质期刊内容一致的网络版, 以单篇或整期出版形式, 在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊 (网络版)》是国家新闻出版广电总局批准的网络连续型出版物 (ISSN 2096-4188, CN 11-6037/Z), 所以签约期刊的网络版上网络首发论文视为正式出版。



基于侧信道特征的安全代理流量分类

高平¹, 广晖², 陈熹¹, 李光松¹

(1. 河南省网络密码技术重点实验室 河南 郑州 450001; 2. 郑州大学体育学院, 河南 郑州 450004)

摘要:近年来,安全代理被越来越多的互联网用户用于规避网络审查和访问受限资源。安全代理流量的分类对于网络安全和网络管理具有重要的意义。为弥补深度包检测技术的不足,提高防火墙流量探测能力,提取了用于安全代理流量分类的侧信道特征,包括有效载荷长度序列、信号序列等。基于这些特征使用机器学习和深度学习算法对 Shadowsocks、V2Ray、Freemate、Ultrasure 4 种被广泛使用的安全代理流量进行识别。实验结果表明,利用与有效载荷内容无关的侧信道特征进行分类,与现有研究相比,在准确率、F1 值等性能方面均有一定程度的提升。

关键词:安全代理;流量分类;机器学习;深度学习;深度包检测

开放科学(资源服务)标志码(OSID):



Traffic Identification

based on Side-channel Features for Security Proxy

GAO Ping¹, GUANG Hui², CHEN Xi¹, LI Guangsong¹

(1. Henan Key Laboratory of Network Cryptography Technology, Zhengzhou, 450001, China;

2. Physical Education College, Zhengzhou University, Zhengzhou 450004, China)

【Abstract】In recent years, security proxy has been utilized by more and more clients to circumvent Internet censorship and access restricted resources. The classification of security proxy traffic is of great significance for network security and network management. In order to make up for the insufficiency of deep packet inspection technology and improve firewall traffic detection capability, the side-channel features used for security proxy traffic classification are extracted, including Payload Length Sequences and Signal Sequence. Based on these features, machine learning and deep learning algorithms are used to identify the widely used security proxy traffic of Shadowsocks, V2Ray, Freemate, and Ultrasure. Compared with the existing research, the experimental result shows that there is a certain degree of improvement in performance such as accuracy and F1-score by using the side-channel features which are not related to the payload content for classification.

【Key words】secure proxy; traffic classification; machine learning; deep learning; deep packet inspection

DOI:10.19678/j.issn.1000-3428.0058954

0 概述

为了维护网络安全,互联网服务商一般会在网关处的防火墙利用网络审查技术来过滤网络流量中的不良信息,如端口过滤、深度包检测等技术。与此同时,一些恶意用户和组织为了规避网络审查往往会使用各种各样的方法手段。目前,用户主要使用安全代理、VPN 加密、匿名网络三种方式突破防火墙的封锁,达到访问受限资源的目的。这些技术对防火墙的穿透会造成用户网络行为无法识别、不可监控的后果,例如非法交易,传播不良信息等。

因此识别和分类这些技术软件产生的流量对网络管理和网络安全具有重要意义。

安全代理部署简单、成本较低,更容易在互联网用户之间传播。已经成为恶意用户突破防火墙的重要手段之一。目前 Shadowsocks(简称 SS)、V2Ray、Freemate、Ultrasure 这 4 种安全代理使用较为广泛^[1]。SS 和 V2Ray 是开源的安全代理软件,用户可轻易在虚拟专用服务器(Virtual Private Server, VPS)上部署。通过 VPS 对客户端流量进行代理,达到穿透防火墙的目的。Freemate 和 Ultrasure 安全代理软件有较大的受众基数,由专门的公司动态网和 UltraReach 对

基金项目:国家重点研发计划项目“网络空间拟态防御技术机制研究”(2016YFB0800100)

作者简介:高平(1996—),男,硕士研究生,主研方向为网络安全;广晖,讲师、硕士;陈熹,讲师、硕士;李光松,副教授、博士。E-mail: yuntshark@163.com

软件进行定期的更新和维护。安全代理通过加密用户的流量来规避防火墙的深度包检测技术。因此安全代理流量分类本质上是加密流量分类问题。加密流量分类主要有三种方法：基于端口的流量分类方法、基于深度包检测的流量分类方法、基于统计特征的机器学习算法流量分类方法。

基于端口的分类是早期传统的流量分类方法，它使用 Internet Assigned Numbers Authority (IANA)^[2] 分配的知名端口号识别流量类型。然而对于使用非标准端口的协议和应用，这种监测技术就会失效。例如 SS 和 V2Ray 安全代理均可指定非标准端口，Freegate 和 Ultrasurf 通过使用 443 端口伪装成 TLS 流量来规避防火墙的端口过滤。

基于深度包检测的分类方法融合了基于端口的分类方法。它不仅在端口过滤网络流量，还会检查流经网关的流量载荷来匹配已知协议或应用的关键字段。目前有 4 种较为流行的深度包检测工具 nDPI^[3]、Libprotoident^[4]、L7 fliter^[5]、OpenDPI^[6]。文献[7]对这些工具的性能进行了比较。然而，深度包检测技术能够识别的协议种类有限，并且对于加密的流量无能为力。由于 SS 和 V2Ray 通过双方预共享口令生成加密密钥，直接加密双方的通信内容，使得流量中分组有效载荷内容被混淆成随机字符串，深度包检测技术不能对其进行过滤和识别。

基于侧信道特征的机器学习算法分类方法通过提取网络流量中分组侧信道特征，例如有效载荷长度、分组时间间隔、分组总数等，使用机器学习算法实现流量类型的区分。相比较基于端口和深度包检测的方法，该方法具有协议无关性，即不依赖有效载荷中协议的具体实现，使用侧信道特征对流量进行分类。

文献[8]基于流中分组总数、有效载荷长度统计量(最大值、最小值、平均值、方差等)和分组时间间隔统计量等特征使用 XGBoost 算法对 SS 流量进行识别，但没有将有效载荷长度直接作为分类特征，并且没有充分利用流中载荷长度的变化。文献[9]基于流中分组总数、同方向分组持续总时间、同方向分组长度总字节数等特征构成 3000 维特征向量，使用随机森林(RF)算法区分 SS 流量。但使用的特征维度高，对于每一条流而言，并不一定具有多达 3000 维特征，特征向量会有大量填充。文献[10]基于流中同方向分组长度总字节数和目的 IP 分布等特征使用随机森林(RF)等机器学习算法区分 SS 流量，但分组时间间隔特征没有被考虑。文献[11]基于流中分组长度平均值、分组总数、分组时间间隔等特征，使用 XGBoost 算法区分 SS 流量，但分组长度平均值并不能精确表征流量的特征变化。文献[12-13]基于分组长度的统计量等特征使用机器学习算法对 SS 安全代理流量进行区分。文献[14]基于载荷内容使用多种深度学习算法区分多种应用，包括 SS。但由于 SS 和 V2Ray 不进行协议的协商，载荷内容被加密算法

加密并直接传输，因此使用载荷内容构造特征向量会对分类造成混淆。文献[15]基于分组长度和分组方向等特征，使用卷积神经网络(CNN)区分 SS、V2Ray 等流量。文献[16]基于分组长度特征识别 Freegate 流量，但由于 Freegate 版本的更新，该方法对更新后的 Freegate 流量并不适用。

相比较于上述文献使用的特征，除了使用流中分组总数、有效载荷长度的统计量、分组时间间隔特征外，流中单个分组的有效载荷长度和流量类型紧密相关，也可以被用来构建特征向量。本文尝试将流中前 n 个有效载荷长度组合成有效载荷长度序列(Payload Length Sequence, *PLS*)，表征流量的载荷长度变化情况。从信号分析的角度，将流中每一个分组表征为一个信号，此信号由有效载荷长度、分组时间间隔、分组方向构成。从而可以将多个分组对应的信号进行合并，构成信号序列，作为分类特征向量。本文基于提取的流量侧信道特征，构建特征向量，使用多种机器学习算法、深度学习算法进行训练得到分类器，最终在测试集上使用分类器区分正常 Web 浏览和 4 种安全代理流量，验证了本文提出特征的有效性，评估了不同算法产生的分类器。使用的分类算法分别包括朴素贝叶斯(NB)、逻辑回归(LR)、支持向量机(SVM)、随机森林(RF)、XGBoost 5 种机器学习算法和前馈神经网络(MLP)、卷积神经网络(CNN)、长短期记忆网络(LSTM) 3 种深度学习算法。

1 预备知识

1.1 安全代理介绍

SS 和 V2Ray 一般由 Socks5 协议代理本地 Web 流量。文献[17]将加密流量通信分为非加密初始化阶段和加密数据传输阶段。非加密初始化阶段协商双方使用的密码算法、DH 值、证书等信息，加密数据传输阶段完成数据的加密传输。但 SS 和 V2Ray 并没有遵循此规则。双方在发送数据之前不会进行协商，而是直接基于双方预共享密钥生成加密密钥，对本地代理和服务器代理之间的数据进行加密封装，如图 1 所示：

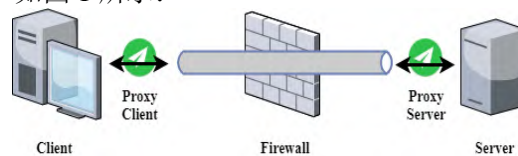


图 1 SS 和 V2Ray 通信流程

Fig.1 Communication process of SS and V2Ray

Freegate 和 Ultrasurf 的工作原理与 SS、V2Ray 不同，不能直接与代理服务器传输数据。首先通过 DNS 解析获取服务器 IP，之后与代理服务器进行协商并建立连接，建立连接后与代理服务器进行数据传输，如图 2 所示：

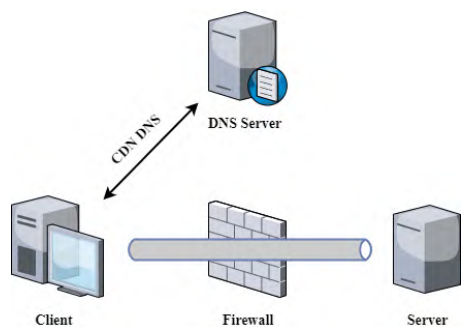


图 2 Freegate 和 Ultrasurf 通信流程

Fig.2 Communication process of Freegate and Ultrasurf

Freegate 有三种通道，分别是 A、F、M 通道。通过测试发现 A 通道和 F 通道已经被防火墙封锁，目前 M 通道可以正常使用。文献 [16]研究了 A 通道。A 通道使用内容分发网络(Content Delivery Network, CDN)隐藏代理服务器真实 IP，使用 TLS 建立连接后使用 UDP 传输数据。与 A 通道不同的是，F 通道使用 TLS 传输数据，但是目前 F 通道不稳定。除了使用 CDN 隐藏代理服务器真实 IP 地址，M 通道建立连接和数据传输均使用 TLS 进行流量伪装，稳定性较好。本文分析 M 通道协商时产生的 TLS 流量。

与 Freegate 的 M 通道相似，Ultrasurf 首先通过 CDN 隐藏代理服务器真实 IP 地址，之后使用 TLS 建立连接并传输数据。

1.2 研究对象

本文以双向流(Bi-flow)为研究对象。设主机 A 的 IP 为 IP_A ，端口为 $Port_A$ ；主机 B 的 IP 为 IP_B ，端口为 $Port_B$ ；传输层协议号为 $Proto$ 。在同一会话中通信双方的 IP 地址、端口号，以及传输层协议号不变并且一直持续到会话结束。从新的会话建立开始，称具有相同五元组 $(IP_A, Port_A, IP_B, Port_B, Proto)$ 或者 $(IP_B, Port_B, IP_A, Port_A, Proto)$ 并持续到会话结束的分组合并为双向流：

$$f = \{h_i p_i\}_i$$

其中 h_i 是流中第 i 个分组的首部，包括以太网层、IP 层、传输层的信息。 p_i 是第 i 个分组的应用层载荷。

2 基于侧信道特征的流量识别方法

由于深度包检测分类方法难以对加密流量有效载荷的关键字段进行匹配，对加密流量的分类性能较差。并且基于 SS 和 V2Ray 在通信时的特点，整个通信过程的数据均被加密算法封装，无协议的非加密初始化阶段，使用有效载荷内容作为分类特征，容易造成混淆。因此，本文提取与有效载荷内容无关的流量侧信道特征，使用机器学习算法和深度学习算法构建分类器，对安全代理流量进行识别和分类。

2.1 主要框架

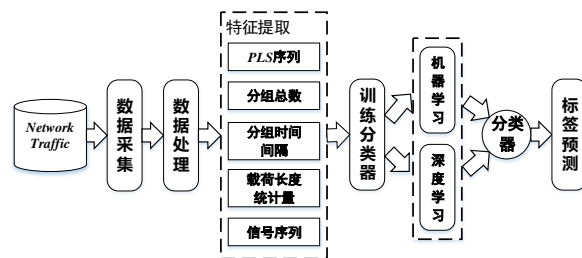


图 3 基于侧信道特征的流量分类框架

Fig.3 Traffic classification framework based on side-channel feature

本文首先收集真实的 Web 浏览和安全代理流量，将流量存储在 pcap 或 pcapng 文件中。之后对流量数据进行预处理，将具有相同五元组的分组组合成双向流(Bi-flow)。提取双向流 PLS 序列、分组总数、分组时间间隔、载荷长度统计量、信号序列侧信道特征，构建特征向量。基于构建的特征向量，使用机器学习和深度学习算法得到流量分类器。之后使用这些分类器在测试集上进行标签预测，并评估不同分类器的性能。本文处理流量和提取特征的工具已在 Github 上公开^[18]。

2.2 特征提取

文献[14][19]使用有效载荷内容，作为特征向量进行流量分类。但是考虑到 SS 和 V2Ray 的流量在通信时没有非加密初始化阶段，整个通信过程被加密算法封装，将有效载荷内容作为分类特征会降低分类器的性能。本文使用 NIST 检测标准说明了 SS 和 V2Ray 的有效载荷内容均被加密，难以进行区分。

NIST^[20]随机性检测标准是美国国家标准与技术研究院(National Institute of Standards and Technology, NIST)提出用来检测二元比特序列随机性的方法集合，其中包含 15 种随机性检测方法。本文使用单比特频数检验、块内频数检验、最大 1 游程检验、近似熵检验、累加和检验 5 种检验方法对 SS 和 V2Ray 流量载荷内容进行分析，使用开源代码^[21]计算检验值，其中块内频数检验、最大 1 游程检验选取的参数如表 1 所示，其他检验使用代码默认参数。

表 1 NIST 标准参数选取

Table 1 The selection of NIST standard parameters		
参数	块内频数检验	最大 1 游程检验
分块大小 M(bits)	20	8

抽取 SS 和 V2Ray 各 50 条流，通过截取流中前 80 字节有效载荷，使用 NIST 随机性检测标准对 SS 和 V2Ray 的通信数据随机性进行度量，表 2 给出了每种检验值的平均值：

表 2 检验标准平均值

Table 2 The average of test standard					
安全代理软件	单比特频数	块内频数	最大 1 游程	近似熵	累加和
SS	0.4895	0.4784	0.5208	0.4999	0.5145
V2Ray	0.4994	0.4724	0.5137	0.5155	0.5418

在 NIST 随机性检测标准中，将检验值与 P 值

比较用来判定测试数据是否为随机产生的。当检验值大于 P 值时测试数据被认为是随机的。本文取 P 值为 0.01, 表示序列被认为是随机序列的置信度为 99%。通过比较, SS 和 V2Ray 流量均为加密流量, 但是载荷的检验值接近, 难以区分。因此需要通过提取和载荷内容无关的并且有区分度的侧信道特征表征安全代理流量, 进而对其进行识别和分类。

定义流 f 中第 i 个分组为 pkt_i , 其有效载荷为 p_i , 有效载荷长度为 $l(p_i)$ 。对于流中前 n 个有效载荷长度不为 0 的分组可定义有效载荷长度序列(Payload Length Sequence, PLS):

$$PLS = (l(p_{i_1}), l(p_{i_2}), \dots, l(p_{i_n}))$$

其中 $l(p_{i_k})$ 表示流中第 i_k 个有效载荷长度不为 0 的分组。根据 n 的不同选取, PLS 的长度也不同。

对于流中前 n 个有效载荷长度不为 0 的分组, 定义第 i_{k-1} 个分组和第 i_k 个分组时间间隔为 t_k , 则分组时间间隔组成的序列如下:

$$T = (t_0, t_1, \dots, t_{n-1})$$

t_0 表示流中第一个有效载荷长度不为 0 的分组和 TCP 三次握手的最后一个分组的时间间隔。

使用有效载荷长度构建 PLS 序列, 可以表征流量载荷长度的变化。为了从整体上对流量进行表征, 流中有效载荷长度统计量包括最大值、最小值、平均值、方差、中位数、上四分位数、下四分位数以及分组总数特征被提取作为特征向量的一部分。

基于上述提取的侧信道特征, 根据算法 1, 构建机器学习算法使用的特征向量 F :

算法 1

输入 $f = \{h_i p_i\}$, 指定 PLS 长度为 n (不包含重传分组), 分组时间间隔序列 T , 分组总数 pkt_cnt , 有效载荷统计量 S , 变量 $k=0$ 。

输出 特征向量 F 。

```

1. For  $pkt_i$  in  $f$  do://处理流  $f$  中第  $i$  个分组
2.   If  $pkt_i$  is retransmitted://判断第  $i$  个分组是
//否为重传分组
3.     continue
4.   Else:
5.     If  $l(p_i)=0$ ://判断第  $i$  个分组有效载荷
//长度是否为 0
6.       continue
7.     Else:
8.        $k++$ 
9.       If  $k \leq n$ :
10.         $PLS.append(l(p_i))$ 
11.         $T.append(t_{k-1})$ 
12.      Else: Break
13. End For
14.  $pkt\_cnt = |f|$ 
15. 计算流中有效载荷长度统计量  $S$ 。

```

16. $F = (PLS, T, pkt_cnt, S)$ //得到特征向量

从信号处理的角度出发, 流 f 可被看作信号源, 流中的分组 pkt_i 可被看作 i 时刻发出的信号 S_i 。 S_i 由分组 pkt_i 的有效载荷长度、分组时间间隔、分组方向构成, 定义如下:

$$S_i = (l(p_i), t_i, dir(pkt_i)) \quad (1)$$

其中 $t_1 = 0$, $dir(pkt_i)$ 定义如下:

$$dir(pkt_i) = \begin{cases} 0 & \text{客户端} \rightarrow \text{服务器} \\ 1 & \text{服务器} \leftarrow \text{客户端} \end{cases} \quad (2)$$

给定要处理流中的分组个数 n , 对每一时刻信号 S_i 进行汇总, 构成流的信号序列特征向量:

$$(2) \quad Sig = (S_1, S_2, \dots, S_n) \quad (3)$$

使用信号序列对网络流量进行表征, 作为 1D-CNN 和 LSTM 深度学习算法的输入, 对安全代理流量进行分类。

2.3 机器学习

本文使用朴素贝叶斯(NB)、逻辑回归(LR)、支持向量机(SVM)、随机森林(RF)、XGBoost 共 5 种机器学习算法对安全代理流量进行分类, 利用算法 1 得到的特征向量 F 组成样本集合进行训练, 并得到分类器。在测试集上, 比较不同 n 值下的不同分类器性能, 最终得到最佳分类器。

对于不同的 n 值, PLS 长度不同。一般来说, 选取流中的分组越多, 分类器的性能越优越, 当选取的分组总数达到一定数量时, 分类器的性能会趋于稳定。

在训练和测试之前首先对数据及进行 Shuffle 操作, 打乱样本顺序。训练集和测试集比例为 3:1。除了 NB 算法, 在训练阶段, 使用 GridSearch 算法进行参数搜索, 并使用 4 折交叉验证。

2.4 深度学习

本文使用 MLP、1D-CNN、LSTM 等 3 种深度学习算法识别和区分安全代理流量。

MLP 架构中的隐藏层如表 3 所示。MLP 的输入特征向量为 $F = (PLS, T, pkt_cnt, S)$ 其中 PLS 和 T 的长度为 8。在每一层隐藏层后加入批量归一化(BN)层, 提高模型的收敛速度, 并防止过拟合。

表 3 MLP 隐藏层架构

层数	每层结构
第一层	128 个神经元+ReLU 激活函数
第二层	BN 层
第三层	64 个神经元+ ReLU 激活函数
第四层	BN 层
第五层	32 个神经元+ ReLU 激活函数
第六层	BN 层

CNN 已经在多个领域内有了较成熟的应用, 例如机器视觉。文献[14][19]基于流中分组有效载荷,

使用 2D-CNN 深度学习算法对加密流量进行分类。根据上文分析, 由于 SS 和 V2Ray 均使用加密算法对数据进行封装, 其流量分组中的载荷均为随机字符串, 将载荷的随机字符串加入到分类特征向量中, 会降低分类器对 SS 和 V2Ray 的分类能力。除了分组有效载荷内容特征以外, 可使用的特征包括有效载荷长度、分组时间间隔、分组方向等流量的测信道特征。对于 2D-CNN 算法而言, 由于卷积核和池化操作均被用来对特征进行降维, 使用侧信道特征构成的特征向量 $F = (PLS, T, pkt_cnt, S)$ 特征维度小, 不适合使用 2D-CNN 进行训练。近年来, 1D-CNN 逐渐被用于语音识别领域[22][23]。将信号序列 $Sig = (S_1, S_2, \dots, S_n)$ 作为 1D-CNN 的输入, 可避免特征维度小而难以进行训练的问题, 其中序列的长度 $n=30$ 。1D-CNN 框架如图 4 所示:

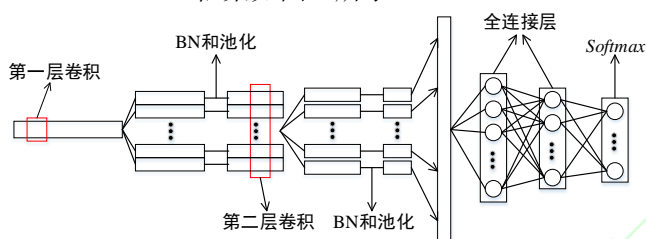


图 4 1D-CNN 整体架构

Fig.4 The overall structure of 1D-CNN

第一次卷积后使用最大池化层, 第二次卷积后使用均值池化层, 每个卷积层和全连接层具体参数如表 4 和表 5 所示:

表 4 卷积层具体参数

Table 4 The specific parameters of convolutional layers

卷积层	大小	卷积核总数	步长	激活函数
第一层卷积	8	128	1	ReLU
第二层卷积	4	64	1	ReLU

表 5 全连接层具体参数

Table 5 The specific parameters of full connection layers

全连接层	神经元总数	激活函数
第一层全连接	128	ReLU
第二层全连接	64	ReLU

LSTM 是专门用来处理时序特征数据的神经网络。将信号序列 $Sig = (S_1, S_2, \dots, S_n)$ 作为模型输入, 对安全代理流量进行分类, 其中 $n=30$ 。其整体架构如下图所示, 其中隐藏层状态为 64, 在 LSTM 隐藏层后加入了一层全连接层, 其神经元个数为 64, 并使用 ReLU 激活函数。最终使用 Softmax 分类器达到分类的目的。

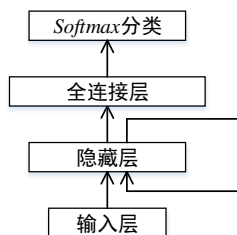


图 5 LSTM 整体架构

Fig.5 The overall structure of LSTM

在训练和测试深度学习算法的分类器时, 首先进行 Shuffle 操作打乱样本顺序, 训练集和测试集比例为 3:1。三种算法的 Batch_Size 大小为 64, 并使用 Adam 优化算法。

3 实验分析与结果

3.1 数据集

通过 Selenium^[24]自动爬取 ChinaZ^[25]排名前 840 的网页, 构造正常的 Web 浏览流量、SS 流量、V2Ray 流量。通过 Bat 脚本自动开启 Freegate、Ultrasurf 代理软件, 构造客户端连接代理服务器的流量。所有的网络流量均存储于 pcap 或 pcapng 文件中, 数据集总量为 10.13GB, 包含 109237 条流。各种类型流量在数据集中具体分布如表 6 所示:

表 6 数据集

Table 6 The dataset

软件名称	版本	操作系统	流总数
Web Browsing	Firefox/IE	Ubuntu 16.04 Windows 10	20793
Shadowsocks	ss-libev 3.3.3 ss-windows 4.1.9.2	Ubuntu 16.04 Windows 10	19141
V2Ray	4.21.3	Ubuntu 16.04 Windows 10	24744
Freegate	7.80	Windows 10	21307
Ultrasurf	20.03	Windows 10	23252

机器学习算法使用 scikit-learn 0.22.2^[26], 深度学习算法使用 Tensorflow 2.0^[27]搭建。

3.2 特征分析

不同的安全代理具有不同的流量特征。从有效载荷长度角度来看, 不同安全代理的有效载荷长度具有不同的分布, 图 6-10 分别列出正常 Web 浏览、SS、V2Ray、Freegate、Ultrasurf 流中第一个分组有效载荷长度的分布。

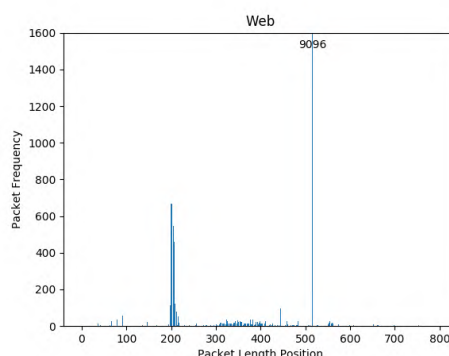


图 6 Web 载荷长度分布

Fig.6 Payload length distribution of Web

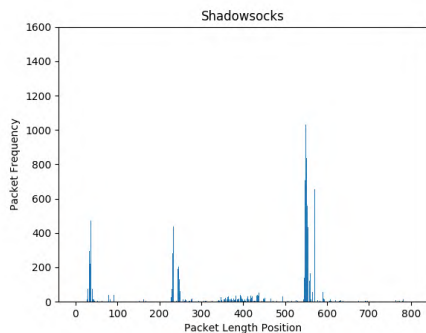


图 7 SS 载荷长度分布
Fig.7 Payload length distrubution of SS

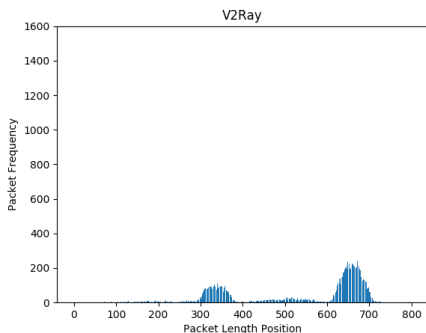


图 8 V2Ray 载荷长度分布
Fig.8 Payload length distrubution of V2Ray

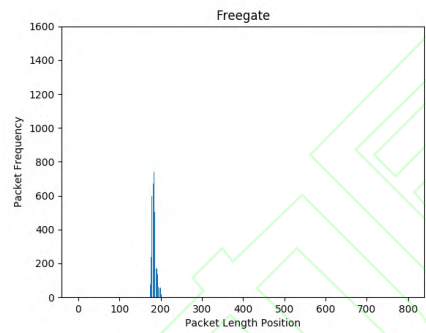


图 9 Freegate 载荷长度分布
Fig.9 Payload length distrubution of Freegate

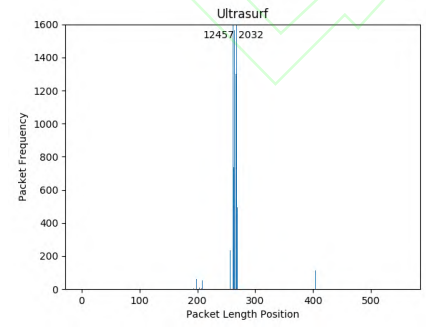


图 10 Ultrasurf 载荷长度分布
Fig.10 Payload length distrubution of Ultrasurf

对于正常的 Web 浏览流量和 SS 流量而言，除了区间[50-100]，对于其他区间的载荷长度分布，SS 的有效载荷长度均大于正常 Web 浏览流量，发生了载荷分布平移的现象。这种现象对 Ubuntu16.04 下的 ss-libev 软件更为明显。通过结合公开的 SS 协议

设计^[28]和源码审计^[29]发现，发生载荷长度平移现象是因为 SS 协议特殊的封装方式造成的。以 ss-libev 为例，SS 协议一般分为 SS 请求和 SS 数据传输，用户访问网站时，首先会构造 SS 请求。SS 请求使用如图 11 所示格式将需要访问的目的地址传递给远端服务器：

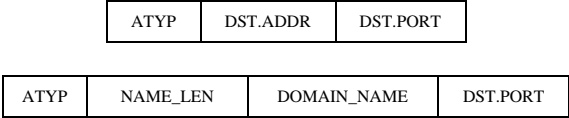


图 11 SS 请求
Fig.11 Request of SS

其中 ATYP 代表地址类型，一般为 IP 和域名两种地址类型。从客户端角度分析，如果访问的网站有其对应的 IP 地址，则使用 IP 地址请求。如果没有，则会将域名填入 SS 请求消息中。客户端等待合适的时机将请求发送给远端服务器，让远端服务器进行地址解析。在发送之前，客户端会对 SS 请求进行加密，如图 12 所示（灰色部分表示数据加密，其中 PADDING 是为了对齐加密算法密钥长度进行的数据填充）：

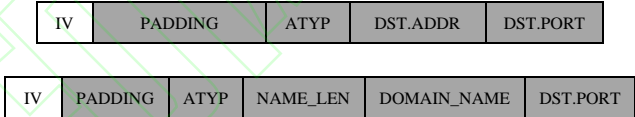


图 12 加密后的 SS 请求
Fig.12 Encrypted request of SS

这时，SS 本地代理通过 Socks5 协议收到浏览器发来的数据，对数据加密封装后和 SS 请求一起发往 SS 服务器。如图 13 所示：

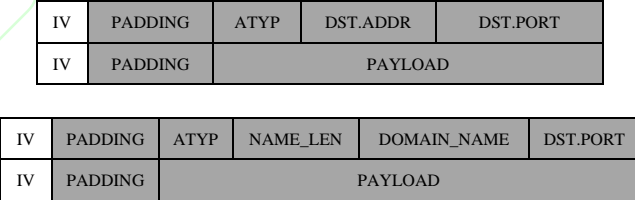


图 13 SS 数据发送
Fig.13 Data transmission of SS

V2Ray 安全代理使用 VMess 协议在客户端和服务端之间进行通信。V2Ray 客户端消息格式如图 14 所示：

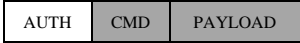


图 14 VMess 客户端消息格式
Fig.14 The client message format of VMess
其中 CMD 的具体格式如图 15 所示：



图 15 CMD 具体格式
Fig.15 Specific format of CMD

IV 为 16 字节，KEY 为 16 字节，Other 字段包括选项、余量、加密方式、指令、端口、地址类型、地址、随机值、校验至少 18 字节。其中余量字段指定了随机值字段的长度，随机值字段代表随机填充的数据。通过随机填充，V2Ray 重塑了所封装流量的分组有效载荷长度分布。因此 V2Ray 载荷长度也会

产生平移现象。

由于 Freegate 和 Ultrasurf 的 TLS 流量中的密码套件和拓展字段与正常的 TLS 流量有差异,因此这些流量的有效载荷长度和正常的 Web 浏览具有一定的区分度。

3.3 实验结果

3.3.1 评价指标

为了评估分类器的性能,本文使用准确率,宏精确率,宏召回率以及宏 F1 值这四个指标。最终给出模型分类结果的混淆矩阵。以上指标的计算如下:

$$Accuracy = \frac{\sum_{i=1}^n TP_i}{N} \quad (4)$$

$$macro-P = \frac{1}{n} \sum_{i=1}^n P_i \quad (5)$$

$$macro-R = \frac{1}{n} \sum_{i=1}^n R_i \quad (6)$$

$$macro-F_1 = \frac{2 \times macro-P \times macro-R}{macro-P + macro-R} \quad (7)$$

其中 TP_i 是每一类正确预测的样本, P_i 是每一类别的精确率, R_i 是每一类别的召回率, N 是测试样本总数, n 为类别总数。

3.3.2 深度包检测

本文首先使用 4 种开源深度包检测工具进行测试,检视这些工具对安全代理流量识别的能力。使用的工具具体信息如表 7 所示:

表 7 深度包检测工具
Table 7 The tools of DPI

DPI 工具	版本	可识别的协议和应用
nDPI	3.0	246
Libprotoident	2.0.12	500
L7 fliter	2020.05.22	127
OpenDPI	3.1.0	90

由于 SS 和 V2Ray 对流量的加密封装,这四种深度包检测工具均不能识别 SS 和 V2Ray 流量。Freegate 和 Ultrasurf 均被识别为 TLS 流量。深度包检测工具主要是通过内置的签名库进行字段匹配,若没有相应的签名匹配则不能识别流量的种类,并且对 Freegate 和 Ultrasurf 的 TLS 伪装流量容易造成误报。因此,防火墙使用深度包检测方法不能有效过滤安全代理流量。

3.3.3 机器学习算法

NB、LR、SVM、RF 以及 XGBoost 5 种机器学习算法以 $F = (PLS, T, pkt_cnt, S)$ 作为特征向量对 4 种安全代理流量进行识别和分类。在不同 PLS 长度下,图 16 和图 17 分别展示了 5 种算法准确率和 F1 值的变化:

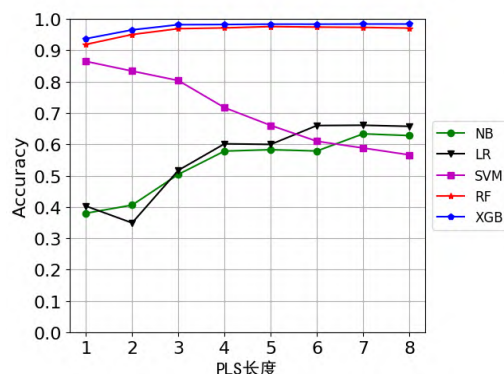


图 16 机器学习算法准确率

Fig.16 The accuracy of ML Algorithms

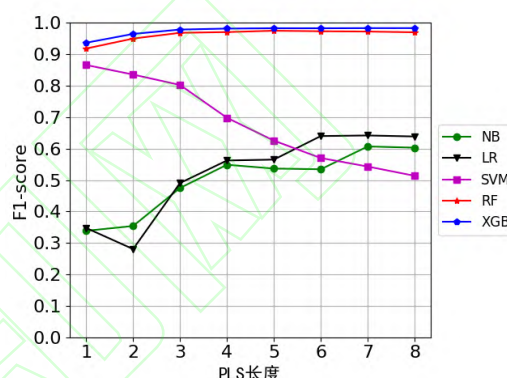


图 17 机器学习算法 F1 值

Fig.17 The F1-score of ML Algorithms

可以看出使用 XGBoost 和 RF 构建的分类器均能够获得良好的分类性能,并且 XGBoost 的准确率比 RF 平均高 1.21%, F1 值平均高 1.2%。当 PLS 长度大于 3 时, XGBoost 和 RF 的分类器性能趋于稳定。当 PLS 长度为 8 时, XGBoost 的准确率为 98.4%, F1 值为 98.36%。相比于 NB、LR、SVM, 使用属于集成学习的 XGBoost 和 RF 算法性能有明显的提升。图 18 给出了 PLS 长度为 8 时, XGBoost 的分类混淆矩阵:

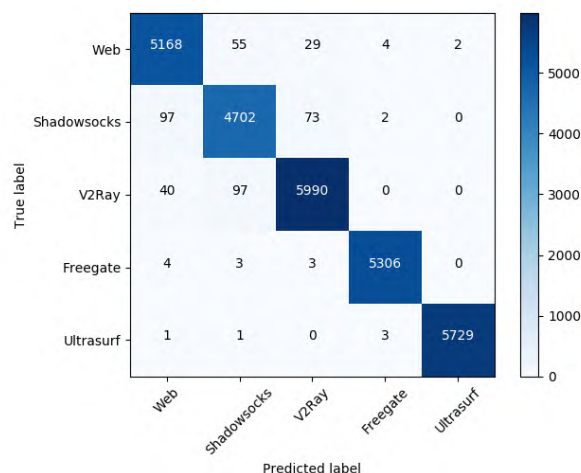


图 18 XGBoost 混淆矩阵

Fig.18 The confusion matrix of XGBoost

3.3.4 深度学习算法

MLP 使用 PLS 长度为 8 时的特征向量 $F = (PLS, T, pkt_cnt, S)$ 作为输入。1D-CNN 和 LSTM 使用构建的信号序列 $Sig = (S_1, S_2, \dots, S_{30})$ 分别对 4 种安全代理流量进行识别和区分。算法的准确率和损失随轮数($Epoch$)增加的变化趋势分别如图 19 和图 20 所示:

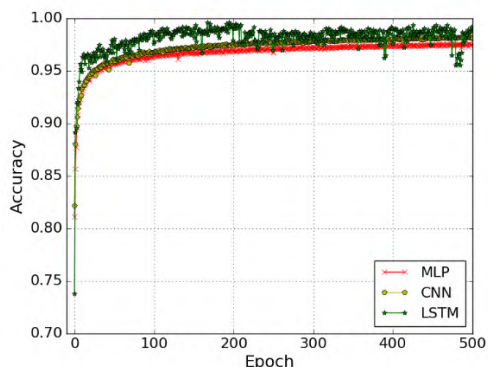


图 19 准确率变化趋势
Fig.19 The trend of accuracy

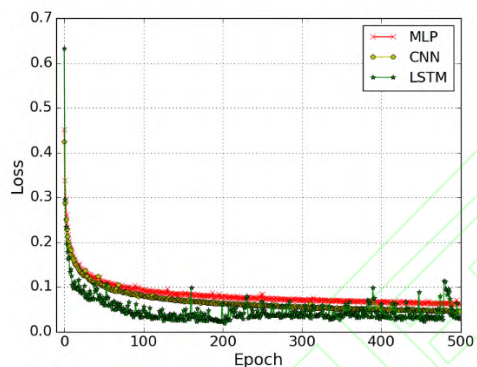


图 20 损失变化趋势
Fig.20 The trend of loss

可以看到,与 MLP 和 LSTM 相比较,1D-CNN 在训练时的准确率和损失随着轮数的增加分别在平稳地增加和减少。最终相比较分类器在测试集上的性能,1D-CNN 的性能最优越。其中 1D-CNN 的分类混淆矩阵如图 21 所示:

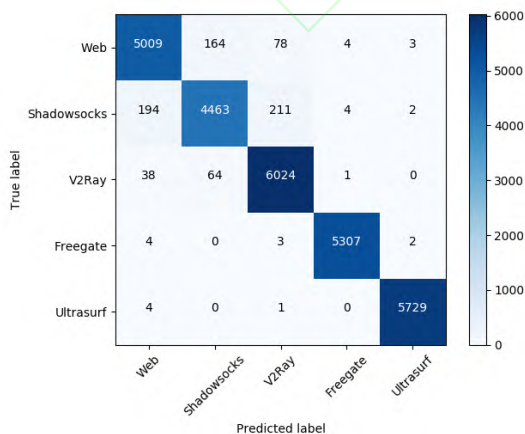


图 21 1D-CNN 混淆矩阵
Fig.21 The confusion matrix of 1D-CNN

3.3.5 结果对比

从分类器性能分析可得出,在 5 种机器学习算法中, XGBoost 性能最优越。在 3 种深度学习算法中 1D-CNN 性能最优越。表 8 对比了 RF、XGBoost、LSTM 以及 1D-CNN 算法在测试集上的性能。

表 8 本文算法性能对比

算法	准确率	精确率	召回率	F1 值
RF	97.10%	96.99%	97.08%	97.03%
XGBoost	98.48%	98.43%	98.45%	98.44%
LSTM	97.02%	96.94%	96.82%	96.88%
1D-CNN	97.15%	97.13%	96.98%	97.04%

从表 8 可分析出 XGBoost 算法的性能是最优的,与 1D-CNN 算法相比较,准确率、精确率、召回率和 F1 值分别提高了 1.33%、1.30%、1.47%、1.40%。

文献[8-12]分别从不同的特征角度对 SS 流量进行区分。除了使用载荷统计量、分组总数表征流量整体的特征,本文提取了反应流量特征变化的 PLS 序列和信号序列,利用序列中特征的不同分布能够获得更好的性能。结果对比如表 9 所示:

表 9 其他文献识别 SS 流量方法性能对比

对比文献	精确率	召回率	F1 值
文献[8]	—	95.14%	—
文献[9]	85.87%	93.28%	89.42%
文献[10]	93.74%	90.66%	92.17%
文献[11]	94.63%	93.28%	93.95%
文献[12]	—	96%	—
本文	96.79%	96.47%	96.63%

文献[15]使用 2D-CNN 对 SS 和 V2Ray 流量进行区分,当训练轮数达到 5000 时,准确率达到 93.3%。相比较之下,本文使用 1D-CNN 在训练轮数达到 500 时,准确率达到 98.22%,计算开销更小。

4 结束语

本文基于流的不同侧信道统计特征包括有效载荷长度序列、分组时间间隔、有效载荷长度统计量、分组总数,构建特征向量对安全代理流量进行表征,分别使用 5 种机器学习算法和 3 种深度学习算法识别和区分安全代理流量。相比较于深度包检测方法,使用机器学习和深度学习算法避免了关键字段匹配技术的缺陷,提高了分类性能。基于本文提取的特征,在本文使用的分类算法中, XGBoost 算法的性能最优越。和其他文献中使用特征获得的分类器性能相比较,例如有效载荷长度统计量、分组时间间隔、分组总数,本文提取与载荷内容无关的侧信道特征通过表征载荷长度变化,区分安全代理流量,达到较好的分类效果,性能有一定提升。下一步将

尝试对其他安全代理软件进行分类,例如Lantern、Trojan、赛风等,加强防火墙的网络审查能力,弥补深度包检测技术的不足。

参考文献

- [1] Callanan C, Dries-Ziekenheiner H, Escudero-Pascual A, et al. Leaping over the firewall: A review of censorship circumvention tools [J]. Freedom House, 2011.
- [2] IANA [EB/OL]. [2020-07-09]. <http://www.iana.org/>
- [3] nDPI [EB/OL]. [2020-07-09]. <https://www.ntop.org/products/deep-packet-inspection/ndpi/>
- [4] Libprotoident [EB/OL]. [2020-07-09]. <https://research.wand.net.nz/software/libprotoident.php>
- [5] L7 filter [EB/OL]. [2020-07-09]. <http://l7-filter.sourceforge.net/>
- [6] OpenDPI [EB/OL]. [2020-07-09]. <https://github.com/thomasbhatia/OpenDPI>
- [7] Bujlow T, Carela-Español V, Barlet-Ros P. Independent comparison of popular DPI tools for traffic classification [J]. Computer Networks, 2015, 76: 75-89.
- [8] HE Hangsong. Research on Shadowsocks Traffic Identification Based on Xgboost Algorithm [J]. Software Guide, 2018, 17(12): 200-203.
何杭松. 基于 Xgboost 算法的 Shadowsocks 流量识别研究[J]. 软件导刊, 2018, 17(12): 200-203.
- [9] Deng Z, Liu Z, Chen Z, et al. The random forest based detection of shadowsock's traffic[C]//2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC). IEEE, 2017, 2: 75-78.
- [10] Zeng X, Chen X, Shao G, et al. Flow context and host behavior based Shadowsocks's traffic identification [J]. IEEE Access, 2019, 7: 41017-41032.
- [11] Cheng J, Li Y, Huang C, et al. ACER: detecting Shadowsocks server based on active probe technology [J]. Journal of Computer Virology and Hacking Techniques, 1-11.
- [12] Han Z H, Chen X S, Zeng X M, et al. Detecting Proxy User Based on Communication Behavior Portrait [J]. The Computer Journal, 2019, 62(12): 1777-1792.
- [13] Aceto G, Ciuonzo D, Montieri A, et al. Multi-classification approaches for classifying mobile app traffic [J]. Journal of Network and Computer Applications, 2018, 103: 131-145.
- [14] Aceto G, Ciuonzo D, Montieri A, et al. Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges [J]. IEEE Transactions on Network and Service Management, 2019, 16(2): 445-458.
- [15] Zhang Y, Chen J, Chen K, et al. Network traffic identification of several open source secure proxy protocols [J]. International Journal of Network Management, 2019: e2090.
- [16] Zhuo Z, Zhang X, Li R, et al. A multi-granularity heuristic-combining approach for censorship circumvention activity identification [J]. Security and Communication Networks, 2016, 9(16): 3178-3189.
- [17] Velan P, Čermák M, Čeleda P, et al. A survey of methods for encrypted traffic classification and analysis [J]. International Journal of Network Management, 2015, 25(5): 355-374.
- [18] Pkt_Analyser Github [EB/OL]. [2020-07-09]. https://github.com/gpsvnc/Pkt_Analyser
- [19] CHEN Xuejiao, WANG Pan, YU Jiahui. CNN based encrypted traffic identification method [J]. Journal of Nanjing University of Posts and Telecommunications (Natural Science), 2018 (6): 8.
陈雪娇, 王攀, 俞家辉. 基于卷积神经网络的加密流量识别方法[J]. 南京邮电大学学报(自然科学版), 2018 (6): 8.
- [20] Rukhin A, Soto J, Nechvatal J, et al. A statistical test suite for random and pseudorandom number generators for cryptographic applications[R]. Booz-allen and hamilton inc mclean va, 2001.
- [21] NIST sp800_22_tests [EB/OL]. [2020-07-09]. https://github.com/dj-on-github/sp800_22_tests
- [22] Oord A, Dieleman S, Zen H, et al. Wavenet: A generative model for raw audio[J]. arXiv preprint arXiv:1609.03499, 2016.
- [23] Aytar Y, Vondrick C, Torralba A. Soundnet: Learning sound representations from unlabeled video[C]//Advances in neural information processing systems. 2016: 892-900.
- [24] Selenium [EB/OL]. [2020-07-09]. <https://www.selenium.dev/>
- [25] ChinaZ [EB/OL]. [2020-07-09]. <https://www.chinaz.com/>
- [26] Scikit-Learn [EB/OL]. [2020-07-09]. <https://scikit-learn.org/stable/>

- [27] Tensorflow [EB/OL]. [2020-07-09].
<https://www.tensorflow.org/?hl=zh-cn>
- [28] Shadowsocks [EB/OL]. [2020-07-09].
<https://shadowsocks.org/en/spec/Protocol.html>
- [29] Shadowsocks Github [EB/OL]. [2020-07-09].
<https://github.com/shadowsocks>

