**西安电子科技大学研究生学位论文**

**撰写要求 (2015年修订版)**

**作者姓名 杨智乐**

**指导教师姓名、职称 苏锐丹 副教授**

**申请学位类别 工学硕士**

**僵尸网络研究**

**作者姓名：**杨智乐

**一级学科：**计算机科学与技术

**二级学科：**信息安全

**学位类别：**工学硕士

**指导教师姓名、职称：**苏锐丹 副教授

**学　　院：**网络与信息安全学院

**提交日期：**2022年3月

**西安电子科技大学**

**硕士学位论文**

**学　号　 19151213547**

**密　级　 公开**

**学校代码 10701**

**分类号** TN82

By

Zhang San

Supervisor: Li Si Title: Professor

February 2015

A thesis submitted to

XIDIAN UNIVERSITY

in partial fulfillment of the requirements

for the degree of selecting one

in selecting one

**Thesis/Dissertation Guide for Postgraduates**

**of XIDIAN UNIVERSITY**

**西安电子科技大学**

**学位论文独创性（或创新性）声明**

秉承学校严谨的学风和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同事对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文若有不实之处，本人承担一切法律责任。

本人签名： 日 期：

**西安电子科技大学**

**关于论文使用授权的说明**

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权属于西安电子科技大学。学校有权保留送交论文的复印件，允许查阅、借阅论文；学校可以公布论文的全部或部分内容，允许采用影印、缩印或其它复制手段保存论文。同时本人保证，结合学位论文研究成果完成的论文、发明专利等成果，署名单位为西安电子科技大学。

保密的学位论文在 年解密后适用本授权书。

本人签名： 导师签名：

日 期： 日 期：

摘要

计算机科学技术发展至今已有40年左右的历史，已经逐渐改变了人们的生活方式。现在计算机已经深刻融入了人们的日常生活之中。其便利性也决定了网络安全问题对人们生活产生的影响是巨大的，小到信息窃取、电信诈骗，大到医疗设备安全问题，不管哪一个都会对人们的财产、安全造成巨大威胁。而网络安全方面规模最大的安全问题，就是僵尸网络问题了。

计算机网络中有一些脆弱的计算机，容易被不怀好意的攻击者攻击并获得该计算机的控制权限，相当规模这样的计算机组织起来，就是恶意攻击者的有力武器。利用僵尸网络，攻击者可以发动分布式拒绝服务攻击，可以发送大量垃圾邮件，可以为互联网灰色领域提供技术支撑。显然，僵尸网络对计算机网络环境造成的危害是巨大的。CNCERT（国家互联网应急响应中心）在07年的《网络安全工作报告》中表明，早在05年，超过5000节点的僵尸网络达到142个之多，最大的僵尸网络节点数目超过了15万个，我国内地被僵尸网络控制的节点数目超过140万。僵尸网络的检测与防御是一个亟待解决的问题。

防范僵尸网络的第一步是检测出正在进行活动的僵尸网络。但是由于僵尸网络控制者也在随着网络安全技术的进步不断地改进僵尸程序，使僵尸网络检测这一任务愈发的难以达到期望的效果。Livadas C[1]等人的研究结果也表明，简单的将机器学习算法应用在僵尸网络流量识别问题上，模型的检测效果十分有限，误报率10%-20%，漏报率30%-40%，不能达到实际应用的准确率要求。 最近几年出现的检测方案，检测重点在于僵尸网络流量数据的特征提取与机器学习算法和深度神经网络在这些特征的分类效果上，除此之外，如何满足大流量数据实时检测的要求也是我们面临的一大问题。

本文论述了僵尸网络检测技术的发展状况，总结了僵尸网络检测方法的一般模式，在此基础上提出了一种组合式的僵尸网络流量特征提取办法，分别提取时间窗口与TCP会话的特征数据，随后使用多种机器学习模型进行训练测试，最终选择了针对时间窗口特征的随机森林模型与针对TCP会话特征的RNN模型共同检测僵尸网络流量。时间窗口是流量检测领域一个常用的识别单位，基于时间窗口的特征提取方案在僵尸网络流量检测领域具有十分广泛的应用。除此之外，TCP会话作为网络应用程序的基本单位，我们可以使用基于TCP会话的特征数据去描述网络应用程序的行为。所以我们选择了时间窗口与TCP流两个维度的特征来提升僵尸网络检测的准确率。模型的构建方面，在对比多个机器学习模型的检测效果后，选择了随机森林与具有处理时间序列能力的RNN神经网络作为模型的分类器，使模型具有学习僵尸程序网络行为的能力。最终结果表明，该方案在CTU-13[2]僵尸网络数据集上达到了97%以上的准确率，在2014年的ISCX-Botnet[3]数据集上也取得了85%的检测准确率，并且可以检测出训练集中未曾出现的僵尸主机。本文详细说明了该方案对比其他检测方案的优势与不足，并在此基础之上提出了未来僵尸网络检测技术可能的发展方向。

**关 键 词**：僵尸网络， 流量识别， 机器学习， TCP会话， 特征选择

ABSTRACT

The computer science has a history of about 40 years, and it has gradually changed people's way of life. Now computers have been deeply integrated into people's daily life. Its convenience also determines that network security issues have a huge impact on people's lives. Information theft, telecommunication fraud, medical equipment security issues, etc., all of which will pose a huge threat to people's property and safety. The biggest security problem in network security is botnets.

There are some vulnerable computers in the computer network, which are easy to be attacked by malicious attackers and gain control permissions of the computers. The organization of such computers on a considerable scale is a powerful weapon for malicious attackers. Using botnets, attackers can launch distributed denial-of-service attacks, send large amounts of spam, and provide technical support for the gray areas of the Internet. Obviously, the harm caused by botnets to the computer network environment is huge. CNCERT (National Internet Emergency Response Center) stated in the 2007 "Network Security Work Report" that as early as 2005, the number of botnets with more than 5,000 nodes reached as many as 142, and the number of the largest botnet nodes exceeded 150,000. The number of nodes controlled by botnets in China exceeds 1.4 million. The form of botnet defense is very serious. Botnet detection and defense is an urgent problem to be solved.

The first step in preventing botnets is to detect an active botnet. However, because botnet controllers are constantly improving bot programs with the advancement of network security technology, the task of botnet detection is increasingly difficult to achieve the desired results. The research results of Livadas C et al. also show that applying machine learning algorithms to the problem of botnet traffic identification cannot meet the accuracy requirements of practical problems (the FPR in the paper is 10%-20%, and the FNR is 30%-40%). The detection schemes that have appeared in recent years focus on the feature extraction of botnet traffic data and the classification effects of machine learning algorithms and deep neural networks on these features, and how to meet the requirements of real-time detection of large traffic data is also a major challenge for us.

In this paper, a combined feature extraction method is proposed, which extracts the feature data of the time window and the TCP stream respectively, and then uses a variety of and its learning models for training and testing. The final results show that the scheme achieves an accuracy of over 98% on the CTU-13[2] botnet dataset, and 88% detection accuracy on the 2014 ISCX-Botnet[3] dataset. Time window is a commonly used identification unit in the field of traffic detection. It is a commonly used traffic identification method to extract traffic features within a time window for model training and prediction. In the field of botnet detection, behavior detection for bots is also a problem that cannot be ignored. TCP stream is the basic unit of network application communication, and a reasonable stream data division method can describe the behavior of the program. Therefore, we choose two dimensions of time window and TCP flow to improve the accuracy of the botnet.

This paper also proposes and implements a real-time data stream parsing method. Combined with the big data processing framework spark, it can meet the requirements of real-time parsing traffic data. The flow data in the time window (only the relevant data packet information is extracted) is divided according to the TCP state machine, and the communication characteristics of the host are analyzed, so as to meet the needs of real-time botnet traffic detection.

**Keywords**: Botnet, Traffic Identification, Machine Learning, TCP Session, Feature Extraction

插图索引

图序号 插图示例 X

表格索引

表格序号 表格示例 X

符号对照表

符号 符号名称

FNR 漏报率

FPR 误报率

XXX XXX

缩略语对照表

缩略语 英文全称 中文对照

FPR XXX XXX

XXX XXX XXX

XXX XXX XXX

目录

[摘要 I](#_Toc97839411)

[ABSTRACT III](#_Toc97839412)

[插图索引 V](#_Toc97839413)

[表格索引 VII](#_Toc97839414)

[符号对照表 IX](#_Toc97839415)

[缩略语对照表 XI](#_Toc97839416)

[第一章 绪论 1](#_Toc97839417)

[第二章 僵尸网络机理 4](#_Toc97839418)

[2.1 什么是僵尸网络 4](#_Toc97839419)

[2.2 僵尸网络的危害 6](#_Toc97839420)

[2.3 僵尸网络的前世今生 7](#_Toc97839421)

[2.3.1 IRC僵尸网络 7](#_Toc97839422)

[2.3.2 P2P僵尸网络 8](#_Toc97839423)

[2.3.3 HTTP僵尸网络 9](#_Toc97839424)

[2.4 僵尸网络的藏匿技术 9](#_Toc97839425)

[2.4.1 Domain-Flux技术 9](#_Toc97839426)

[2.4.2 Fast-Flux技术 10](#_Toc97839427)

[第三章 检测方案研究 12](#_Toc97839428)

[3.1 核心难点 12](#_Toc97839429)

[3.2 检测方法 13](#_Toc97839430)

[3.2.1 基于内容特征 13](#_Toc97839431)

[3.2.2 基于行为特征 14](#_Toc97839432)

[3.2.3 基于增殖网络攻击 15](#_Toc97839433)

[3.2.4 基于多维数据的检测系统 15](#_Toc97839434)

[第四章 特征提取方法 16](#_Toc97839435)

[4.1 数据集 17](#_Toc97839436)

[4.2 基础特征 18](#_Toc97839437)

[4.3 基于时间窗口 19](#_Toc97839438)

[4.4 基于TCPSession 21](#_Toc97839439)

[4.4.1 TCPSession提取方法 22](#_Toc97839440)

[4.4.2 TCPSession特征提取 29](#_Toc97839441)

[第五章 基于时间窗口与TCPSession的RNN分类器 32](#_Toc97839442)

[5.1 循环神经网络 32](#_Toc97839443)

[5.2 检测模型 35](#_Toc97839444)

[5.3 模型评估 39](#_Toc97839445)

[第六章 总结与展望 43](#_Toc97839446)

[参考文献 53](#_Toc97839447)

[致谢 57](#_Toc97839448)

[作者简介 59](#_Toc97839449)

# 绪论

随着计算机科学与技术在人们日常生活中的应用，人们依赖计算机网络完成日常生活的程度日益加深。互联网作为信息互联互通的基础设施，与人们日常生活中的方方面面相结合，可以极大程度的简化办事流程，提高社会运转效率，比如快捷支付、网购平台、基础服务的信息化，都方便了我们的日常生活。但是，互联网与我们的生活结合如此深刻，也决定了互联网的安全问题会对我们的日常生活造成严重影响。比如隐私问题，在我们依赖电子设备提供的便利性的同时，我们的信息也被电子设备接入的云端服务掌握，不仅如此，还存在一些恶意的软件工具主动地窃取电子设备的隐私信息；网络钓鱼问题，攻击者伪造正常网站骗取用户凭证，利用窃取的用户凭证执行非法操作；各种网络漏洞，攻击者利用信息系统或语言框架自身的漏洞执行危险操作，窃取信息等等。在过去的二十年中，网络安全领域发展迅速，各种攻击手段层出不穷，随之而来的是大面积的机器沦陷，造成重大经济损失。

计算机网络中存在一些脆弱的计算机，容易被恶意攻击者攻击并获取控制权限，许多这样被控制的计算机组织起来，就形成了攻击者实施恶意行为的有力武器。其中，攻击计算机获取控制权限的软件被称为僵尸程序，被该程序攻陷的主机被称为僵尸主机（Bot），许许多的的僵尸主机连接起来，组成了僵尸网络（Botnet），僵尸网路有一个被称为BotMaster的控制者，控制僵尸网络整体的行为。BotMaster可以控制僵尸网络发动各种网络攻击，比如DDOS（分布式拒绝服务攻击）、SPAM（垃圾邮件）、PHISHING（网络钓鱼）、CLICK FRAUD（点击欺诈）等等[4]。

僵尸网络不仅用于发动传统网络攻击，而且会被用于模拟真人在社交网络上的活动。恶意操控者通过虚拟账号在社交平台发布信息影响舆论，会对网络舆论环境造成恶劣影响。Mesnards N G等人在2016年美国大选数据集上的研究表明，只要保证充足的活跃度，哪怕实际数量还不到参与用户总数的1%，Twitter上的机器人账号，依旧能够影响舆论的风向走势[5]。Twitter、Facebook、QQ、微信、微博是现在社交平台的代表，如果其中的僵尸账号在关键的选举时间给出错误的误导信息影响舆论，造成的威胁将是巨大的。

僵尸网络可以给控制者提供可控制的网络资源（IP地址、计算资源），IP地址资源可以作为代理隐藏真实的主机，公共的IP地址还可以用来架设违法网站以牟取利益，恶意的攻击者可以通过为互联网黑色产业（电信诈骗等）提供支持的方式，比如在境外为诈骗、钓鱼、色情、赌博等违法犯罪网站提供网络服务，获取经济利益，对网络环境造成了及其恶劣的影响。

僵尸网络对网络环境的影响非常恶劣，抑制僵尸网络的野蛮生长是网络安全领域的重要课题。但是，僵尸网络的检测与防治确并不是一项简单的任务。一方面僵尸网络复杂多样，难以给出统一的检测方案，并且随着攻击者安全意识的提高，恶意软件网络通信普遍使用了TLS加密，为我们的检测制造了困难，另一方面，恶意攻击者针对相应的僵尸网络防御技术不断地优化自身恶意工具与僵尸网络的结构，使得恶意软件与僵尸网络的行为更加的难以检测与预防。

协议方面，僵尸网络一开始使用IRC协议，比如1999年的PrettyPark僵尸网络[6]。IRC（Internet Relay Chat）协议本来用于网上聊天，但其工作机制完美契合了僵尸网络控制者控制僵尸网络的需求。但IRC协议容易被检测系统识别，随着网站数量的指数级增长，HTTP协议得到广泛的应用。与IRC协议相比，HTTP协议具有通用性更强、更加难以检测的特点，所以越来越多的僵尸网络开始使用HTTP协议来作为命令与控制信道的协议，比如2004年的Bobax[6]僵尸网络。然而使用HTTP协议构建的僵尸网络依然具有中心化的结构，控制服务器被捣毁会导致僵尸网络的瘫痪，为了克服中心化结构僵尸网络控制中心脆弱的缺点，黑客们使用P2P协议构建出分布式僵尸网络，比如2002年的Slapper[6]僵尸网络。与通信协议相对应的，僵尸网络结构也从中心化结构发展到了稳定性更好的分布式结构或局部中心化，整体分布式的混合结构[7]。

僵尸网络的检测技术，也经历过了迭代发展的过程。人工智能技术应用在网络安全领域已经被证明是构建检测系统的有效方法。僵尸网络的行为隐藏在各种各样的数据之中，在检测之前我们需要先从原始数据中提取特征，在这些特征的基础上，应用机器学习领域的工具，以帮助我们完成僵尸网络检测的任务。提取特征我们可以从网络流量分析、协议特征、增值网络攻击[7]这三个方向来入手。网络流量分析方面，因为僵尸节点的网络流量具有时空相似性，可以根据这一点检测僵尸节点。协议特征方面，有些通信协议的特征十分明显，比如IRC僵尸网络中通过某种硬编码的算法生成具有规律的昵称信息，导致IRC客户端具有相似昵称，可以根据这一点检测IRC僵尸网络。增值网络攻击方面，通过检测僵尸网络发动的攻击行为来检测僵尸网络，比如通过识别DDOS与大规模的垃圾邮件，来检测僵尸网络。在特征的基础上，检测方法又分基于签名的检测和基于行为的检测

基于签名的检测是基于DPI（深度数据包探测）技术，检测数据包中是否出现了恶意通信流量的特征。这种检测方案一般要借助IDS的检测能力，提前配置好僵尸网络流量的签名信息，当发现数据包出现了与签名信息库中相同或相近的内容时，就检测到了僵尸网络流量。这种方案的优点是不容易产生误报，准确率较高。但是缺点也很明显，DPI技术需要识别数据包内容，在面对大流量的检测需求的时候，显得有些力不从心，对比每个数据包的payload内容与特征库中每条特征，计算量巨大，基于签名的检测技术难以处理大规模流量数据。而且基于签名的检测技术的基础，是签名（僵尸网络流量提取到的特征），如果僵尸程序使用加密过的数据通信，那么基于签名的检测技术将无法成功检测到恶意通信流量。针对新出现的，特征库中未曾记录的僵尸网络，基于签名的检测技术也无能为力。

基于行为的检测是针对僵尸程序的行为与普通程序的行为差异，来实现僵尸网络检测。虽说是基于行为，但是在计算机的世界里，描述程序的行为靠的还是各种各样的指标数据，从恶意通信流量中获取可以描述程序行为的特征，根据这些特征建立行为模型，随后使用这些行为模型去匹配从待检测流量提取的特征，以此来实现恶意僵尸网络流量的识别。比如Zamboni等人根据僵尸节点网络通信流量时空相似性原理提出的检测方法TAMD[8]。

目前虽然有各种各样的检测方案，但是在检测算法的应用过程中存在以下问题：

1. 准确率有待提高  
   僵尸网络检测方案的一大重点是方案正确识别僵尸网络的能力和从背景流量中找出僵尸网络流量的能力。目前的检测方案的准确率虽然在论文中使用对应的数据集可以达到工程应用的水平，但是实际应用时区分僵尸网络流量的能力并不像论文描述的那样出色。
2. 实时过滤流量的能力有待提高  
   僵尸网络检测方案部署的设备是网关，每秒流过的数据可能有数百G之多，如何实时的处理如此庞大流量数据的计算任务是一个非常有挑战性的难题。

本文提出了一种基于时间窗口与TCPSession上下文的僵尸网络检测方案，该方案提取时间窗口内流量数据的统计特征，以及若干个TCPSession上下文的流统计特征，一同作为判定僵尸网络流量的依据。时间窗口是流量识别领域常用的检测单位，可以通过计算某个时间窗口的网络流量的统计数据来完成流量识别任务，比如Binkley等人在2006年提出的基于TCP扫描权重的僵尸网络检测方法[9]。同时，TCPSession作为描述主机网络行为的基本单位，在主机的行为检测上也是一个重要的基本单元，比如W Wang等人在2020年提出的基于流与基于图的僵尸网络检测方案[11]就使用了TCP会话作为提取特征的基本单位。所以，本文选择了基于时间窗口与基于TCP会话的特征来实现僵尸网络检测。

本文第二章主要描述了僵尸网络的运行机理与发展过程，与之对应的检测方案的发展过程，概述了僵尸网络检测的关键难点，现有检测方案，以及现有检测方案的优势与不足。第三章主要介绍了实时的数据流处理方案，如何从流数据中实时的提取TCPSession，如何快速的实现时间窗口流量统计特征的提取，并且使用传统机器学习分类器检测了基于TCP会话的特征与基于时间窗口特征在僵尸网络数据集与正常网络流量数据集的分类效果。第四章叙述了本文提出的僵尸网络方案的内部原理与技术细节，介绍了基于时间窗口的随机森林分类器与基于TCPSession特征的循环神经网络（RNN）分类器的僵尸网络检测方案，并使用ISCX-Botnet数据集检验了模型的实际效果。第五章分析了本文提出的检测方案的结果，指出了一些该方案仍然存在的不足。第六章总结了本文提出的检测方案，阐述了僵尸网络技术和针对僵尸网络的检测技术未来可能的发展方向。

# 僵尸网络机理

## 什么是僵尸网络

在互联网中，有一些安全防护能力弱的主机，被恶意攻击者通过恶意软件入侵的方式获取控制权限，许许多多这样的主机连接起来组成的行为协同的计算机网络，组成了僵尸网络（Botnet）拓扑。攻击者（BotMaster）可以通过命令与控制信道向僵尸网络节点发送指令，控制僵尸网络节点发动DDos攻击或发送垃圾邮件等恶意活动，由于僵尸网络节点众多，规模巨大，其恶意活动造成的影响也十分恶劣。广义的僵尸网络包括了僵尸网络节点、命令与控制信道、僵尸程序、僵尸网络控制者。

僵尸网络节点是被攻击并获取控制权限的主机，是直观的组成僵尸网络的主体。僵尸网络的规模也直观表现在僵尸网络节点的数量，大型的僵尸网络节点数目可以到万级别，CNCERT（国家互联网应急响应中心）在07年的《网络安全工作报告》中表明，早在05年，超过5000节点的僵尸网络达到142个之多，最大的僵尸网络节点数目超过了15万个，我国内地被僵尸网络控制的节点数目超过140万。如果这些主机联合起来开展恶意活动，其造成的影响将是十分巨大的。

僵尸程序（Bot）是可以攻击计算机获取控制权限，并且可以主动加入僵尸网络并具有自传播行为的程序。僵尸程序是僵尸网络扩张并赖以生存的工具，以恶意软件的形式存在于互联网。这类程序通过计算机网络服务的漏洞，或社会工程学方法（垃圾邮件）入侵宿主计算机，获取控制权限，接受僵尸网络控制者（BotMaster）发送的行为指令，并在合适的时间展开新的传播行为。僵尸程序是维持僵尸网络生命的重要部分，是网络安全问题的规模化表现。

命令与控制信道（Command & Control Channel）是BotMaster控制僵尸网络的方式、协议。僵尸网络控制者通过命令与控制信道向僵尸网络注入指令，僵尸网络节点接收到指令后开展恶意活动。命令与控制信道是僵尸网络的核心，在中心结构的僵尸网络中，如果C&C通信被容易的检测出来，防御人员可以通过捣毁控制中心的方式轻易的捣毁僵尸网络。所以，僵尸网络命令与控制信道的健壮性、安全性是僵尸网络健壮性与安全性的基石。初代的僵尸网络使用IRC协议作为C&C通信的协议，IRC（Internet Relay Chat）协议用于因特网聊天，运行IRC客户端协议的主机连接到某个频道之后，可以接收到来自该频道的所有信息。这一特性和僵尸网络控制者组织僵尸网络、发布僵尸网络指令的需求不谋而合。但是IRC协议特征明显、恶意主机使用IRC协议与BotMaster通信的过程容易被检测出来，所以恶意攻击者采用了应用更加广泛的HTTP协议。HTTP协议是无状态协议，需要Bot定期的轮询BotMaster获取控制指令，与IRC协议相比，HTTP协议具有应用广泛，内容模式难识别的优点。随着检测技术的进步，中心化结构僵尸网络变得更加难以生存，催生了使用P2P协议的分布式僵尸网络。P2P协议没有服务器与客户端之分，每个主机都是对等的，可以将单个主机的计算、带宽资源整合到整个网络之中，一般用于网络资源下载。P2P协议应用在僵尸网络通信中，主要为了隐藏真实的BotMaster，去除了中心化僵尸网络中心节点脆弱的弊端。僵尸网络使用P2P协议与HTTP协议相比更加安全，难以检测，但是通信效率不如IRC与HTTP协议。命令与控制信道在发展过程中还出现了一些类似Domain-flux与Fast-flux的有趣技术，一定程度上增加了中心化结构僵尸网络的健壮性 。

BotMaster是僵尸网络控制者，既指的是控制中心服务器，也指的是通过C&C信道向僵尸网络注入恶意指令的攻击者。

攻击者通过各种方式使目标主机感染僵尸程序，随后僵尸程序使用Rootkit等技术隐藏自身，使得僵尸程序可以不被系统防护工具清除。僵尸程序通过某种方式寻找其余感染节点并加入僵尸网络，比如在IRC僵尸网络中，新感染的节点会连接指定的IRC服务器，通过加入指定频道的方式来加入僵尸网络。僵尸程序在受害计算机被攻陷后等待接收指令采取进一步行动，当接收到BotMaster传来的攻击指令时，即可发动Ddos攻击或发送垃圾邮件。与此同时，运行在目标主机上的僵尸程序也会以目标主机为基础不断地感染新机器传播自身。掌握了0day漏洞的僵尸网络传播速度往往是指数级别的，而且在传播之初较难反制。比如出现在2019年7月的Moobot僵尸网络，利用Telnet弱口令或0day漏洞传播自身，发动不间断的Ddos攻击。

僵尸网络是计算机网络安全问题的规模化表现，是有史以来最大规模几次DDOS攻击的罪魁祸首，不仅如此，僵尸网络还互联网黑色产业提供网络资源支撑，影响恶劣。僵尸网络会造成重大网络安全问题，如何有效防御与反制僵尸网络是网络安全领域的热点问题。

## 僵尸网络的危害

僵尸网络在BotMaster的控制下，可以统一协作，对特定目标发起攻击，其中一种攻击形式就是DDOS。Mirai僵尸网络诞生于2016年[12]，主要感染目标是运行linux操作系统的物联网设备。Mirai僵尸网络从2016年8月开始活跃，在开始活动的第二天，Mirai已经感染了65000个IoT（Internet Of Things）设备，传播速度非常快，在早期传播阶段，其规模每76分钟扩大一倍，在11月份达到峰值600000。 Krebs是一位报道网络犯罪的独立记者，由于其独特的工作方向，许多犯罪分子经常向其博客发动DDOS攻击。从2012年7月至2016年9月期间该博客遭受了269次DDOS攻击，而Maria僵尸网络攻击是规模最大的一次，攻击流量达到了623Gbps，甚至迫使其CDN服务商停止了Kerbs博客的DDOS防护服务。不仅如此，Mirai针对欧洲云服务提供商OVH的攻击活动流量高达1Tbps，规模十分惊人。僵尸网络发动DDOS攻击会对网络环境造成恶劣影响，类似Mirai的IoT僵尸网络的兴起，促进了DDOS攻击商品化的进程[13]，同时也警示了安全工作者，物联网设备的安全性正随着物联网技术的广泛应用变得愈发重要。

统一协作的僵尸网络不仅可以发动DDos攻击，还可以大规模发送垃圾邮件。垃圾邮件传播不良信息，并且一般以营利为目的配合黑客开展钓鱼活动，还有些垃圾邮件以传播恶意软件为目的，是僵尸网络的重要传播扩展方式。著名的垃圾邮件僵尸网络Grum每天发送的垃圾邮件数目高达400亿条，占总垃圾邮件的四分之一，虽然在2011年被捣毁了，但是该僵尸网络造成的影响十分深远。

运行在宿主机上的僵尸程序，可以记录用户输入获取用户登录凭证，获取宿主计算机的敏感信息；可以在宿主计算机安装广告插件以牟取利益；可以以宿主机为基础更大范围的传播自身。僵尸网络对网络的威胁不仅威胁了个人在网络中的虚拟资产，而且在社交网络中活跃的僵尸账号，甚至可以影响舆论的走向。Mesnards N G等人在2016年美国大选数据集上的研究表明，只要保证充足的活跃度，哪怕实际数量还不到参与用户总数的1%，Twitter上的机器人账号，依旧能够影响舆论的风向走势[5]。研究人员从不会改变立场的顽固用户中，锁定了396个僵尸账户，不到样本总数的5%，虽然占比不高，但是僵尸账号的活跃度远超真人，这些僵尸账户将双方候选人的观点推向极端，吸引了数量可观的真实支持者，使得两位候选人在Twitter平台的支持度实现了微妙的平分秋色。上述案例说明，无论是不是SNS（Social Network Service）网红，只要投入足够的资本发送SNS僵尸网络营造声势，最终的舆论走向，可能和公众期望的结果大相径庭。美国大选尚可如此，那么资本驱动的SNS营销活动，也是如此[14]。无论是对传统计算机网络，还是对社交网络服务，僵尸网络造成的影响都是十分恶劣的。

运行在PC主机上的僵尸网络对互联网造成的影响是巨大的，而且最近10年间僵尸网络也有像互联网的其他领域扩展的迹象。移动互联网设备与物联网设备是僵尸网络的重点扩展对象。随着移动互联网的发展，智能手机在人们生活中的地位越来越高，如果出现了针对智能手机的僵尸程序，其传播速度与感染规模将会上一个台阶。由于Android系统的开放性，搭载Android系统的手机成为僵尸程序的主要目标设备。僵尸程序可以伪装成第三方软件，引诱用户下载，获取用户敏感信息，并在必要的时候响应中心服务器发动网络攻击。物联网技术由于其便利人们日常生活的特性，正在向我们生活的各个方面渗透，随处可见的无人饮料售卖机，工厂的各种监控设备，农业中的各种监控设备等等，为我们的生活带来便利的同时，也因为自身安全性问题引入了安全隐患。物联网设备被攻破之后会直接的影响到这些设备的正常功能，造成经济损失，并且响应攻击指令，发动DDOS等网络攻击。2016年出现的Maria网络开了IOT僵尸网络的先河，从此IOT僵尸程序层出不穷，并且都具有不俗的生存能力，造成的危害也是巨大的。

## 僵尸网络的前世今生

僵尸网络从出现开始，便一直随着网络安全技术的进步不断改进自身。命令与控制机制与僵尸网络节点的组织结构是僵尸网络之间的主要差别，命令与控制机制决定了僵尸网络使用何种通信协议，使用通信协议的不同进一步影响了僵尸网络节点的组织结构。根据C&C信道使用协议的不同僵尸网络可以分为IRC僵尸网络、HTTP僵尸网络、P2P僵尸网络。

### IRC僵尸网络

IRC僵尸网络使用中心化结构，每个僵尸主机都需要连接IRC服务器以接收BotMaster的指令，当IRC服务器被捣毁时，整个僵尸网络将无法继续工作，具有单点失效的弊端，而且IRC协议的协议特征明显，容易被检测系统识别，但是其通信协议简单，与P2P僵尸网络相比通信高效，僵尸节点可以在很短的时间内完成对BotMaster的响应。

使用IRC协议的僵尸网络是出现最早的僵尸网络。IRC僵尸网络的诞生最早可以追溯到1993年12月份的Eggdrop[15]。Eggdrop是一个为Unix用户编写的IRC聊天程序，在编写之初是为了方便用户管理与保护聊天室的聊天内容，但是由于其管理IRC聊天内容的方式与僵尸网络管理者（BotMaster）管理僵尸网络节点的需求不谋而合，有众多的黑客受Eggdrop的启发开发了第一代僵尸程序，比如1999年出现的PrettyPark，该蠕虫给IRC僵尸网络技术奠定了十分重要的基础，它包含的一些功能直到现在仍然在广泛使用，比如获取主机信息，搜索用户密码等敏感信息，自我更新，上传、下载文件，发动拒绝服务攻击等。在2000年时GT-Bot出现，该僵尸程序具有端口扫描、泛洪攻击、复制并主动传播自身的功能。2002年出现的SD-Bot，标志着僵尸网络传播技术的另一个里程碑，SD-Bot传播自身的欲望十分强烈，SD-Bot会不断扫描其他主机，用尽各种办法（NetBios、Rpc服务、各种后门）进行攻击，一旦得手便会在目标主机下载安装SD-Bot，完成自身的传播行为。后续出现的IRC僵尸网络都继承了这一特点。同样是2002年出现的僵尸网络，AgoBot在软件设计上引入了模块化，使得AgoBot僵尸程序的功能扩展变得十分容易，至此，IRC僵尸网络技术走向成熟。

### P2P僵尸网络

僵尸网络在发展的同时也很快引起了安全人员的注意，在攻防对抗的过程中，攻击者逐渐发现了使用IRC协议作为命令与控制信道的通信协议容易被识别，为了克服中心化结构僵尸网络容易被反制的缺点，P2P协议成为了构建僵尸网络命令与控制信道的有效协议。P2P的核心在点对点，每个节点既充当服务器又充当客户机，即使部分节点被检测出来，整体僵尸网络的行为也不会受到太大影响。但是P2P协议的通信效率不如IRC，P2P网络中的僵尸主机无法在短时间内接收到BotMaster发出的行动指令，有一些P2P僵尸网络引入了中心节点来提高网络的通信效率或更好的管理僵尸网络，但同时也引入了单点失效的缺点。

P2P僵尸网络的不同主要在于发现其他被感染节点的方式不同。在2002年出现了第一个使用P2P协议的僵尸网络Slapper[16]，在该僵尸网络中，每个节点都建立了一个完整的已感染主机列表，使得Slapper的通信效率较为高效，但是当僵尸节点数量很多，已感染主机的列表会变得难以维护，而且Slapper没有认证技术，导致该僵尸网络容易被劫持。2003年出现的Sinit，新感染的僵尸节点则采用随机扫描的方式发现其他已感染主机，这就导致了Sinit僵尸网络的节点连接度不高，而且节点在寻找“组织”的时候会产生大量的扫描行为，变得容易被识别检测。2004年出现了使用WASTE协议的P2P僵尸程序Phatbot，虽然WASTE协议本身没有中央节点，但是Phatbot选用了Gnutella发现僵尸节点，扩展性有限。2006年出现的Nugache僵尸程序，使用预定义的22个中心节点发现僵尸节点，当中心节点故障时，整个僵尸网络将无法继续增长。2007年基于Kademlia协议的完全分布式僵尸网络Storm的出现，标志着P2P僵尸网络技术的成熟。2008年出现的Waledac僵尸网络，混合使用了多种协议以保证僵尸网络健壮性，也标志着P2P僵尸网络进入了一个新的阶段。

### HTTP僵尸网络

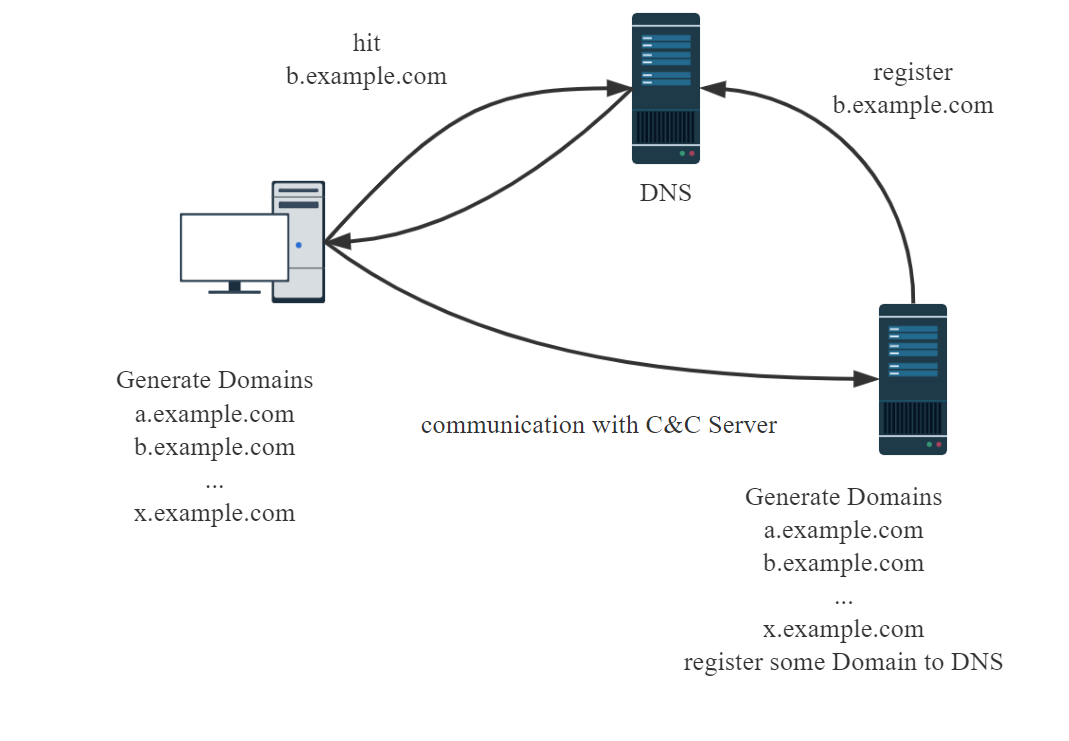
HTTP协议是Web的通信基石，我们使用HTTP协议才能访问数以亿计的网站。HTTP协议是浏览器与web服务器之间的通信协议，应用十分广泛。HTTP协议可以用来传输网页数据，传输媒体数据，传输用户认证凭证，传输命令数据，并且可以以多种不同的方式传递这些信息，模式复杂多样，应用十分广泛。

由于web站点众多，HTTP协议虽然特征明显，但是很难区分一个HTTP包携带的内容是异常的还是正常的，尤其是使用了TLS加密的情况下，对僵尸网络来说，HTTP协议具有灵活、隐蔽的优点。使用HTTP协议构建僵尸网络，仍然具有中心化结构的特点，但是其健壮性要优于IRC僵尸网络。采用HTTP作为通信协议的僵尸网络有Rustock, Clickbot, Naz, Zeus, Conficker, Torpig。

## 僵尸网络的藏匿技术

### Domain-Flux技术

Domain-Flux技术用来解决中心化结构僵尸网络中心节点容易被发现的问题，是一项古老但是活跃的技术，给打击和关闭使用该技术的僵尸网络带来了不小的麻烦。Domain-Flux技术可以达到的效果是，让僵尸网络中心控制服务器的域名随时间发生改变，给防护人员定位恶意域名增加阻碍，以此来实现中心控制服务器的隐藏。其原理如图1所示



图表 1 Domain-Flux原理示意图

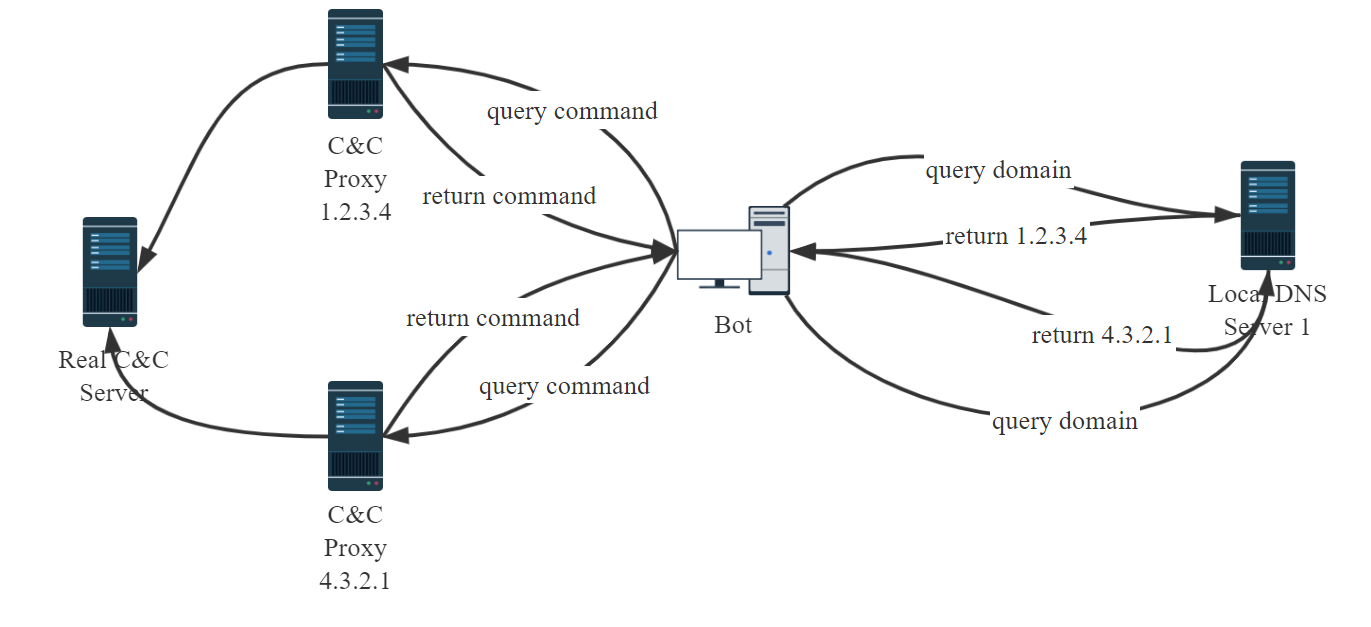
提起Domain-Flux，就不得不说DGA（Domain Generation Algorithm）域名生成算法。DGA算法可以根据输入动态生成包含大量域名的集合。僵尸网络控制服务器与僵尸节点使用相同的DGA算法生成域名集合，僵尸网络控制服务器选择若干个真实使用的域名，将其注册到DNS服务器。僵尸节点寻找控制节点时，遍历生成的域名，直到找到可以访问的域名，就找到了控制中心。DNS注册信息的更改需要配合这些有效域名的变化，如果频繁的向DNS服务商提交域名变更请求，会容易被DNS服务商识别，但是攻击者如果控制本地DNS服务器，就可以让这些更改变得更加随意且不易被察觉。第一个使用此技术的已知恶意软件家族是2008年的Kraken。同年晚些时候，Conflicker蠕虫将DGA的策略推向了恶名。 即使经过10年，仍然可以在当今的某些网络上找到Conflicker或其变体之一。

### Fast-Flux技术

Fast-Flux技术也是僵尸网络常用的用于隐藏自身控制中心的方法。Fast-Flux通过频繁更改DNS（控制DNS服务器）、使用反向代理的方式，达到控制中心IP地址不断变化的的目的。僵尸网络有许许多多被攻击者控制的主机，这些主机上可以运行代理程序将发往这些主机的请求转发到真实的C&C服务器，形成一个具有众多代理节点的反向代理网络。在我们日常使用网络的情况下，用户针对某个域名的DNS查询的结果在很长一段时间都是不会发生变化的，但是Fast-Flux可以频繁的改变域名与IP地址之间的映射，在短时间内查询通过Fast-Flux技术部署的域名，会得到不同的结果。根据实现技术的不同，Fast-Flux又分为Single-Flux与DoubleFlux。

#### Single-Flux

Single-Flux是指只有一层变化的Fast-Flux，攻击者控制底层的域名服务器，频繁修改目标域名对应的IP地址，同时使针对目标域名DNS查询结果的有效时长变短，达到客户机查询同一域名得到不同IP地址的效果。其解析过程如图2 所示

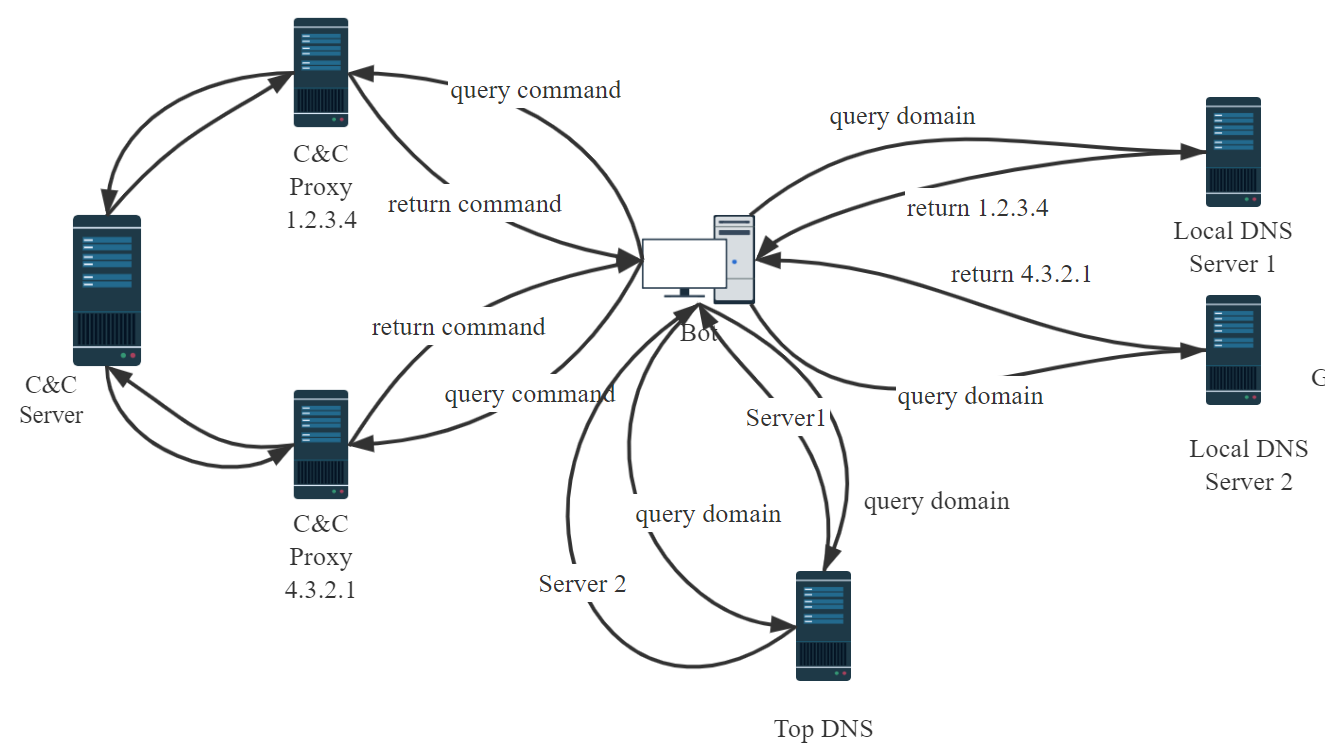


图表 2 Single-Flux原理示意图

僵尸主机查询被控制的DNS主机，DNS主机返回时效短的IP地址，僵尸主机访问该地址获取控制指令，由于DNS查询时效短，先前查询得到的IP地址很快失效，导致僵尸主机产生新的DNS查询，得到另一主机的IP地址。需要注意的是，僵尸网络控制者需要控制底层的DNS域名服务器，因为频繁修改域名IP地址的映射对DNS服务提供商来说是异常行为，如果有对本地DNS服务器的控制权限，修改便不再受到限制。

#### Double-Flux

Double-Flux在Single-Flux的基础上增加了一层变化。攻击者控制多个底层DNS服务器，给每个服务器针对目标域名配置不同的IP地址，并不断修改顶级域名服务器中，对应的底层域名服务器的IP地址。当僵尸主机通过DNS查询域名对应的IP地址时，顶级域名服务器返回含有该域名的底层DNS服务器（被攻击者控制），僵尸主机查询该DNS获取控制主机的地址，该过程如图3所示。



图表 3 Double-Flux原理示意图

正常来说，僵尸网络控制者是不可能有顶级域名服务器的控制权限的，为了避免僵尸网络暴露，Double-Flux的修改频率比Single-Flux要低很多。

# 检测方案研究

## 核心难点

检测僵尸网络时可以获取的数据有限。检测僵尸网络不比维护内网安全性，在内网中我们可以从应用、防火墙、主机、网络等多个维度共同组织防护手段，全方位的阻止恶意入侵的发生。但是僵尸网络检测系统部署的位置往往是边界网关，目标是通过检测识别流经网关的流量数据识别出僵尸网络主机。这样的网络环境中，我们是无法为我们无法控制的主机部署防护系统的，这决定了我们僵尸网络检测系统可以接触的数据边界，就是流经网关的流量数据。当然如果我们要检测的主机在内网中，我们还可以统一记录主机的日志，从日志中也可以识别主机中应用的行为，以此来识别类似僵尸网络的恶意软件。

僵尸网络技术一直处于发展之中。僵尸网络技术在过去的二十年中也在随着网络安全技术的进步不断地改进自身，从一开始的集中式IRC僵尸网络到现在的分层或分布式的P2P僵尸网络，其复杂程度增加了好几个数量级。一些僵尸网络还使用TLS对自己的连接数据进行加密，有些中心化结构的僵尸网络还是用Domain-Flux与Fast-Flux技术隐藏自己真实的命令与控制服务器，这些都为我们识别与反制僵尸网络增加了困难。

检测僵尸网络的特征难以选择。检测僵尸网络的难点在于，从僵尸网络流量数据中提取何种特征，使用何种判定规则来使僵尸网络流量与正常网络背景流量具有更高的区分度。Livadas[18]等人应用机器学习领域内的多种分类器去识别IRC僵尸网络流量，实验结果表明，从流量数据中区分IRC流量的误报率为2.49%， 漏报率为15.04%， 从IRC流量中区分正常流量与僵尸网络流量误报率达到了10%-20%， 漏报率30%-40%。以上案例表明，简单的将机器学习算法应用在僵尸网络检测的问题上无法达到我们预期的检测效果。僵尸网络检测一直是网络安全领域的热点话题，相关的论文也一直像雨后春笋。有许多作者提出了不同的复杂分类器来实现僵尸网络流量与正常流量的分类，不过笔者认为，僵尸网络检测更重要的是从僵尸网络的工作机制、协议特征、僵尸节点的行为等方面来思考，选取那些特征可以让僵尸网络流量相对于正常流量更容易的区分。当选取的特征使用所有的机器学习分类器和神经网络都无法有效的进行识别任务时，就要考虑我们选取特征的合理性了。目前也有一些检测人员提出了一些特征提取方案，只是在论文的数据集上取得了很好的效果，但是在实际应用中的检测效果有待考证。

不仅如此，僵尸网络种类繁多，不同结构的僵尸网络在通信协议、僵尸网络行为上有很大的不同，比如分布式P2P僵尸网络与集中式的HTTP僵尸网络，针对单个僵尸网络的检测方案难以推广到其他的僵尸网络实例，也为我们指定检测方案增加了难度。

即便检测僵尸网络是一项困难的任务，但是也有一些优秀的安全人员提出了一些优秀的检测方案，成为僵尸网络检测领域的重要参考。

## 检测方法

### 基于内容特征

有些特征明显并且未使用TLS协议加密的僵尸网络，可以通过内容比对的方式检测，如早期采用特定TCP与UDP端口的Nugache与Sinit僵尸网络。由于IRC协议中客户端订阅频道时需要提供自己的昵称，而僵尸网络主机的昵称一般由程序生成，具有模式相似的特点。Goebel[24]等在07年提出了一种基于这种原理的检测方案，该方案使用ngrep工具从网络流量中提取IRC协议，评定IRC连接中的异常昵称信息，检测出被IRC僵尸网络感染的主机。陈连栋[27]等人在2016年提出了一种检测分布式僵尸网络的方法。该方案提出了主机关系链，主机网络流图谱的概念，通过学习不同僵尸程序的通信行为，从这些通信中解析主机关系链与网络流图谱，提取特征后，建立检测模型来识别相似流量从而实现僵尸网络流量的检测。

基于内容特征的检测技术检测效率较高，不容易产生误报，而且部署成本较低，应用较为广泛。但是内容特征基于某种特定的僵尸程序，如果该僵尸程序使用了加密算法或逃避技术，内容特征将不再有效。而且此种类型的检测系统对尚未提取过特征的僵尸网络也无能为力。除此之外，大量僵尸程序的特征提取工作也是一个很有挑战性的任务，虽然目前也出现了一些自动化提取僵尸程序特征的方法，但是其可用性还有待考证。

### 基于行为特征

基于行为特征的僵尸网络检测方法就是通过识别僵尸程序和正常程序之间的行为差异，来实现僵尸网络检测。Yen[19]等人在2008年根据僵尸网络多个主机的通信流量具有时空相似性这一特点，提出了一种检测方案TAMD。该方案从三个方面定义流量相似性：相同地址、相同平台、类似内容。基于僵尸主机网络通信时空相似性的检测方案还有BotSniffer[21]和BotMiner[22]。BotSniffer根据僵尸程序与命令与控制服务器通信具有时间与模式关联性的特点，从网络的IRC与HTTP流量中提取可以流量，再结合一些恶意事件的日志，来判定僵尸主机与C&C服务器。BotMiner尝试将流量数据按照目的地址与端口进行聚合，分析得到通信行为相似的组，再结合异常事件的日志来判定僵尸主机。僵尸网络流量的行为特征也会表现在僵尸程序使用的网络服务上，比如DNS查询记录，Choi[20]等人在2007年提出了一种根据异常DNS查询记录来检测僵尸网络的方法，僵尸程序查询命令与控制服务器IP地址的行为具有时间上的关联性，该方案将DNS查询目标域名记录下来，如果在不同的时间段之内，针对这些域名的DNS查询的源IP地址是固定的，则这些主机有可能是僵尸主机，该域名有可能是命令与控制服务器的域名。

僵尸程序的行为特征还体现在网络流量的统计特征上。僵尸程序与正常程序的流量差异也体现在流量数据的统计特征上，在合适的特征集合上应用机器学习算法与神经网络分类器，可以达到分类僵尸网络流量的目标。Strayer[25]等人在2006年提出了一种基于TCP扫描权重的僵尸网络检测方法。该方案基于僵尸程序会扫描其余主机并发起网络攻击的特点，检测主机的扫描行为，来实现僵尸网络的检测。邹福泰[26]等人在2021年提出了一种基于生成对抗网络的僵尸网络检测方法。该方案提取网络数据流，提取时间维度的统计特征与空间维度的流量图像特征（将网络流量看做图片），随后使用生成对抗网络生成僵尸网络特征样本，最后使用双向LSTM分类器完成僵尸网络检测任务。李可[28]在2017年提出了一种基于HTTP一问一答的检测方法。该方案选择了HTTP通信的首个请求应答包组合，提取该包的统计特征，构建分类器实现HTTP僵尸网络流量的识别。

TCP数据流作为网络主机通信行为的基本单位，在刻画程序网络行为上具有重要的作用，在TCP数据流上提取出的各种特征，比如平均每个包字节数，平均每个数据流的字节数，平均每秒的字节数等等，也可以成为我们检测僵尸网络流量的重要依据，基于数据流统计特征的检测方案一般具有检测多种僵尸网络的能力，但是不具备对检测未出现的僵尸网络的能力。最近几年出现了许多基于数据流特征检测僵尸网络的方案。Kirubavathi[31]等人在2016年提出了一种基于网络数据流统计特征的僵尸网络检测方案，该方案提取固定时间间隔内的数据流的统计特征，使用这些特征构建分类模型，实现僵尸网络检测。Zhao[32]等人在2013年提出了一种基于时间窗口数据流统计特征分析的僵尸网络检测方案。该方案选取了12个数据流上的统计特征，可以描述时间窗口内的行为特征，随后使用决策树算法完成数据流的分类。该方案可以检测未知僵尸网络，并在僵尸程序的活动的早期检测到僵尸程序与C&C服务器的通信。Alauthaman等人在2018年提出了一种基于决策树与深度神经网络的检测P2P僵尸网络的方法。该方案首先提取必要的流量数据，选择数据流上的29个特征，随后使用特征选择算法CART去除影响较小的特征，随后建立分类模型完成僵尸网络流量的检测。该方案达到了99.2%的检测准确率。

基于行为异常的僵尸网络检测方案不容易被僵尸程序绕过，并且由于僵尸程序固定的工作模式，这种类型的检测方案针对位置类型的僵尸网络也有一定的效果，不过目前基于行为异常的检测方案取得的成效并不令人满意，主要是因为僵尸程序的行为复杂多样，互联网背景流量的行为模式同样复杂多样，难以统一的定义异常行为模式，可以将二者很好的区分开来。此外，基于行为异常的检测方案需要经过复杂的计算，难以满足实时处理流量数据的要求。

### 基于增殖网络攻击

僵尸网络在传播阶段，会使用各种手段扫描目标主机，尝试利用网络漏洞入侵目标主机获取控制权限，随后下载僵尸程序完成僵尸网络的扩张。僵尸网络在攻击阶段多个主机会一起发动对某个目标地址的攻击行为，所以僵尸网络的行为往往和一些恶意网络事件在时间上关联性较大，并且依然具有群体相似性的特点。可以针对这两点来指定僵尸网络检测方案。Zhuang[23]等人在08年提出一种基于垃圾邮件内容相似性来识别僵尸网络的方法，将垃圾邮件根据内容进行分组，再分析发送者IP来识别僵尸主机。

### 基于多维数据的检测系统

僵尸网络流量隐藏在正常流量之中，其模式复杂多样，攻击者还会使用各种各样的技术躲避检测，比如使用TLS协议对通信内容进行加密，使用Domain-Flux与Fast-Flux技术隐藏C&C服务器等，所以要完成僵尸网络检测的任务，只提取网络流量数据进行分析是不能满足我们的要求的。僵尸程序的活动主要有扫描、漏洞利用、恶意代码注入、连接C&C，可以考虑从这些方面识别僵尸程序。

BotHunter[29]是一个整合多维度信息的僵尸网络检测系统。BotHunter基于证据链思想，将僵尸程序的活动视为一系列有联系的关键信息序列，包括扫描、漏洞利用、命令与控制信道的连接等步骤。BotHunter结合了Snort入侵检测系统，使用了Snort的特征检测方法，并使用了Snort中的两个关键插件用以检测恶意代码攻击与扫描事件。BotHunter将IDS给出的报警信息联系在一起，给出一个完整的僵尸程序“犯罪现场”。BotHunter首次提出了一个刻画僵尸程序整个活动过程的分析系统，是后来研究者设计僵尸网络检测系统的重要参考。

Almutairi[30]等人在2020年提出了一种基于主机与网络的混合僵尸网络检测方法。该方案基于主机和网络两个维度的信息，实现了在僵尸网络的早期阶段检测僵尸网络的效果，并且具有检测新型僵尸网络的能力。该方案提取了主机端文件系统与注册表的信息，网络流量端僵尸程序的传播流量与僵尸程序和C&C通信的流量信息，聚合这两个维度的数据，使用HAHABot[30]算法检测僵尸网络通信。该方案的优点是可以检测出尚未出现的僵尸网络，在除了论文中提及的其他数据集上也有很好的分类效果。但是该方案需要在主机端安装额外的防护监控软件，配合网络端数据一起分析僵尸网络流量，部署要求较高。

# 特征提取方法

近10年间，机器学习算法与深度神经网络正在成为我们解决网络安全问题的重要工具。安全领域的异常检测、入侵检测，都可以抽象为二分类问题，比如异常指令检测、入侵检测、恶意软件检测等，都可以使用机器学习算法和神经网络来解决。僵尸程序通信行为复杂多样，我们使用传统的基于规则引擎的方式过滤网络流量难以涵盖所有的僵尸程序，而且僵尸程序的活动呈现一定的上下文关联性，简单的抽取一小部分流量难以判断该流量是否正常。所以，简单的将机器学习算法工具应用到僵尸网络识别的问题上不能取得好的检测效果。使用机器学习的方法来解决安全问题时，最重要的是安全问题的抽象，合适特征的选取。安全问题抽象是指将我们实际遇到的，网络安全问题抽象成基于数据的分类问题，合适的特征选取要求我们熟悉安全问题的内在机制，从数据中抽取合适的部分输入决策工具得到我们想要的分类结果。在僵尸网络检测问题上，我们部署在网关的检测系统可以触及到的数据，就是流经网关的流量数据，这一现实背景决定了我们识别僵尸网络主机的数据界限，被限定在了网络流量之中。

如何从单一的流量数据中识别僵尸网络通信数据是我们面临的难题，简单的应用算法工具无法解决问题，因为僵尸网络的行为被隐藏在通信数据之中，而我们没有一种很好的方式从流量数据中来刻画僵尸程序的行为。时间窗口是我们在流量识别领域常用的提取单位，提取某个时间窗口内某个主机的通信流量，随后选择若干特征，使用这些特征来训练模型，完成流量识别任务。（使用时间窗口进行流量识别任务的实例）。TCPSession是一个TCP连接中收发双方的所有数据包的有序集合，是计算机程序网络活动的基本单位，从TCPSession相关的数据之中，可以窥探到程序的部分行为。 （使用TCPSession进行流量识别任务的实例）本章节提出了一种僵尸网络流量的特征提取方案，该方案同时提取时间窗口上的流量特征与TCPSession上下文的流量特征，从两个维度刻画程序的网络行为，为检测模型的建立提供基础数据。

## 数据集

数据集的选择上，考虑三个因素：足够的样本、僵尸网络实例丰富、背景流量类型丰富，从符合这些特征的数据集中提取到的特征，才更能体现出真实网络环境中网络流量的状况。

CTU-13数据集是捷克理工大学在2011年捕获的数据集。该数据集旨在构建一个真实的僵尸网络流量与正常的背景流量相混合的完整数据集合。该数据集包含了13个不同的恶意僵尸程序的活动场景，解析数据包并将数据包重组为双向数据流，提取了部分TCP流的特征，方便使用。CTU-13数据集包括了Neris、Rbot等僵尸程序，涵盖了DDos攻击、垃圾邮件、僵尸程序启动、fast-flux、UDP scan等恶意场景，在该数据集上可以取得有效检测效果比在一些含有僵尸网络数据较为单一的数据集上取得的检测效果更有说服力。

ISCX-VPN[34]数据集是纽布伦斯威克大学网络安全研究所创建的包含日常常见流量类型的使用VPN与未使用VPN的数据集合。该数据集包括了14中不同类型的流量数据，包括VOIP、VPN-VOIP、P2P、VPN-P2P、浏览器数据（Firefox、Chrome）、电子邮件、媒体流数据等等。该数据集可以在一定程度上代表正常运行的网络中正常的背景流量。

ISCX-Botnet数据集是纽布伦斯威克大学网络安全研究所创建的僵尸网络数据集，该数据集合并了多个知名数据集，将恶意网络流量映射到本地与正常网络通信背景流量混合，克服了传统僵尸网络数据集僵尸网络流量恶意样本单一的问题。该数据集共包括16种不同类型的僵尸网络流量和未知的背景流量，恶意样本丰富，并且网络通信流量更贴合实际，用来训练与检测僵尸网络检测系统的成果十分合适。

我们选择了已打标签的包含多个僵尸网络数据的CTU-13数据集作为我们的恶意样本，选择ISCX-VPN数据集作为我们的正常样本，从这两个数据集中提取特征数据训练并测试模型，随后在ISCX-Botnet-2014[3]数据集应用我们的检测模型，检验我们模型的效果。我们提取时间窗口和TCP流上下文两个维度的特征数据，两个维度共同的特征在4.2节描述，时间窗口特征在4.3节描述，基于TCPSession的特征提取方法和提取的特征内容在4.4节描述。

## 基础特征

有一些特征在时间窗口维度和TCPSession同时有效。这些特征如表1所示

表格 1 公共特征表

|  |  |  |
| --- | --- | --- |
| 特征 | 描述 | 用途 |
| Protocol | 传输层协议 | ----- |
| PX | 总的传输数据包个数 | 行为相似性 |
| NNP | TCP标志位为NULL的数据包个数 | 协议特征 |
| PSP | 小数据包占比 | 行为相似性 |
| IOPR | 传入数据包与传出数据包的比例 | 行为相似性 |
| RCC | 重连次数 | C&C连接 |
| TBT | 总的传输字节数 | 行为相似性 |
| APL | 每个数据包平均负载字节数 | 行为相似性 |
| DPL | 相同长度的数据包数量占比 | 行为相似性 |
| PV | 数据包长度的标准差 | 行为相似性 |
| ABPS | 每秒的平均字节数 | 行为相似性 |
| PS | 每秒平均数据包个数 | 行为相似性 |
| FR | 连接失败的比率 | C&C连接 |
| DPL | 数据包大小的数量/数据包个数 | 行为相似性 |
| MPL | 最大数据包的尺寸 | 行为相似性 |
| MP | 最大数据包的个数 | 行为相似性 |

这些特征被同时用在基于时间窗口与TCPSession的特征提取阶段，目标是为了根据协议特征、僵尸网络通信行为相似性和C&C信道特征来识别僵尸网络流量。

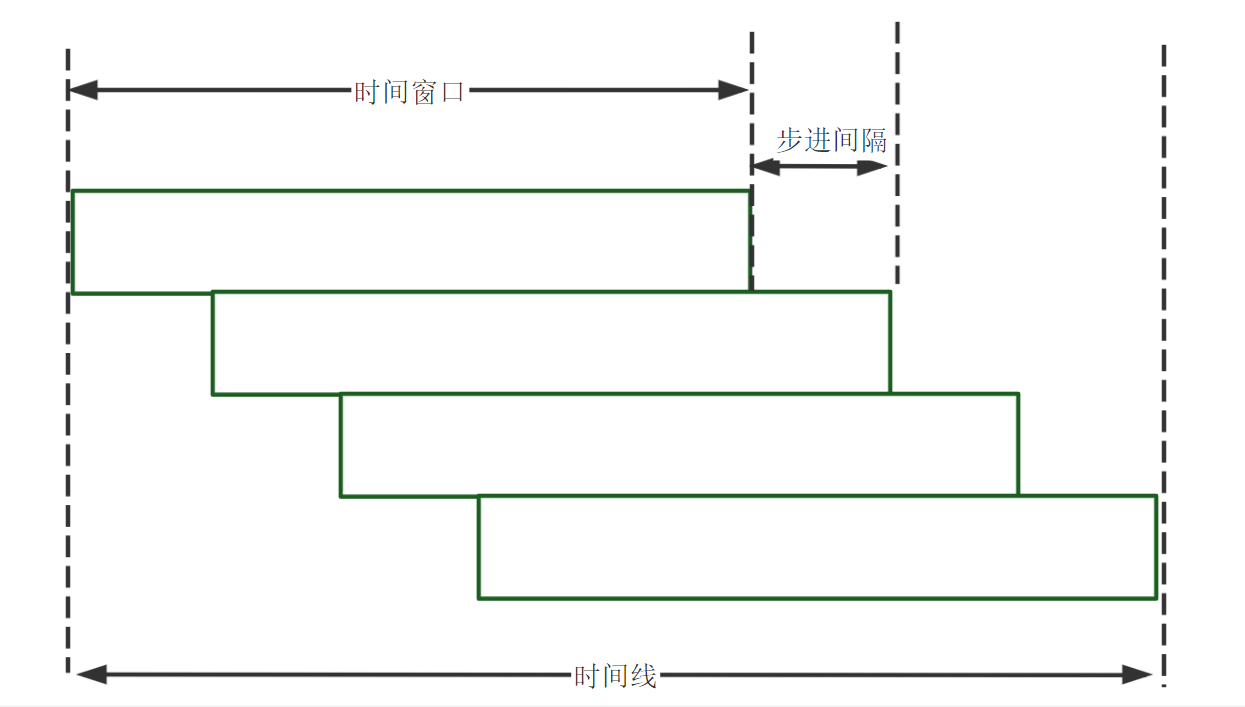
## 基于时间窗口

基于时间窗口的特征如表2所示

表格 2 基于时间窗口的特征

|  |  |  |
| --- | --- | --- |
| **特征名称** | **含义** | **用途** |
| TTF | 在该时间窗口中TCP流的个数 | 行为相似性 |
| TTUC | 在该时间窗口中唯一连接（TCP四元组）的个数 | 行为相似性 |
| TCC | 某个主机尝试发起连接的次数 | 建立C&C连接 |
| UDPR | 某个主机尝试连接的目的地址（二元组）的个数 | 建立C&C连接 |
| FR | 连接失败的比例 | 建立C&C连接 |
| CPR | 控制数据包的比例 | 协议特征 |
| RCPR | 接收的控制数据包的比例 | 协议特征 |
| TCPR | 发送的控制数据包的比例 | 协议特征 |
| UR | UDP协议包所占比例 | 协议特征、DNS查询比例 |

时间窗口是流量识别领域常用的检测单位，将时间窗口内的流量收集起来，提取目标特征后输入检测模型做检测，以识别流量类型或检测入侵行为。Alauthman[35]等人在2020年提出了一种基于强化学习的僵尸网络检测方案，该方案中使用了滑动窗口的方式，使用步进时间窗口提取特征数据来完成流量检测任务，如图所示



图表 4 步进时间窗口原理示意图

我们选择合适的时间窗口提取我们需要的流量统计特征，随后将该时间窗口向前移动固定的时长，继续统计该窗口内流量的统计特征，这样得到的特征更具有实时性，且不会有流量内容被时间窗口分割导致检测效果不好的弊端。

时间窗口大小与步进间隔大小的选择是很重要的因素，对我们的检测结果影响较大。过大的时间窗口会将某个时间的数据特征分散平均到整个时间区间，使得原本较为突出的特征在时间窗口上取了平均值之后变得平常，对模型来说难以区分，过小的时间窗口又会导致特征提取不完整，难以完整描述僵尸程序的通信行为。步进间隔越小，我们提取的特征越多，实时性越好，但是我们需要的计算也更繁重，如何找到合适的时间窗口大小与步进间隔是我们的重要目标。我们实现了时间窗口与步进间隔可变的特征提取方案，通过修改时间窗口大小与步进间隔来找到最合适的参数组合。

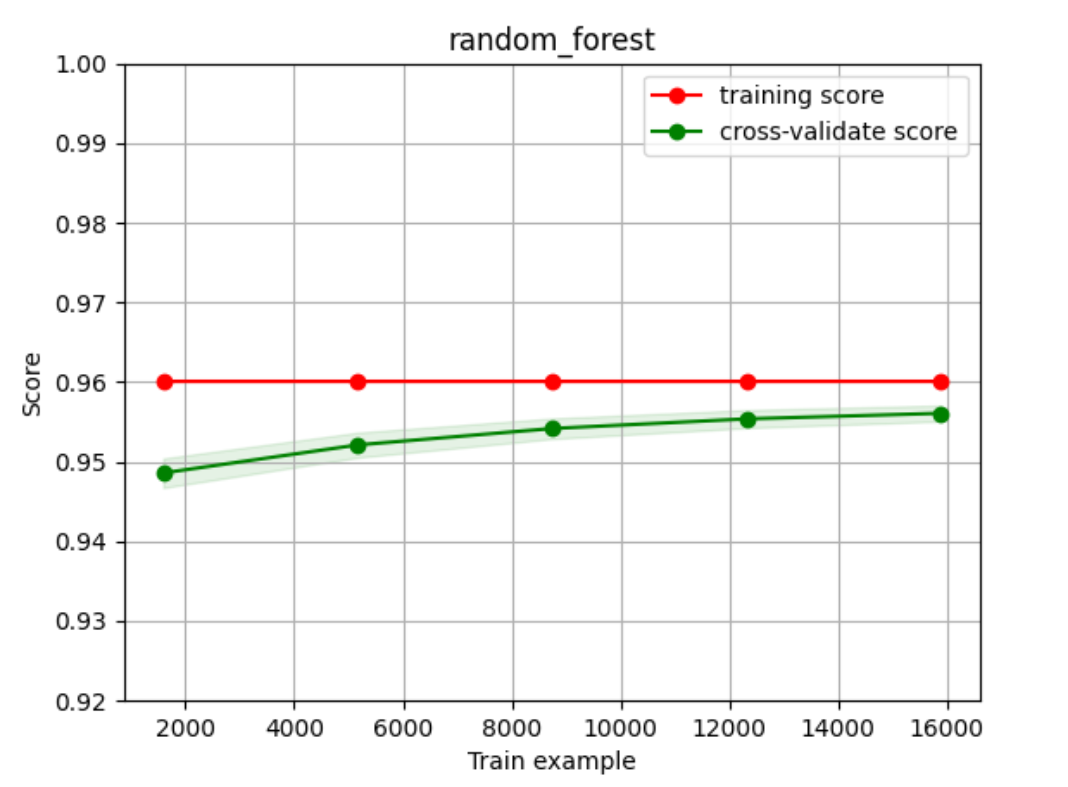
我们使用传统的机器学习分类器测试了我们的特征提取方案在ISCX-VPN数据集与CTU-13僵尸网络数据集上的区分度。在我们的实验数据集上，选择时间窗口大小为10s，步进间隔为5s，使用贝叶斯分类器、线性回归分类器、支持向量机分类器、决策树分类器和随机森林分类器建立分类模型，在该数据集上取得的结果如表3所示

表格 3 各个分类器的分类效果

|  |  |  |  |
| --- | --- | --- | --- |
| 分类器 | 查准率 | 召回率 | F1 Score |
| 贝叶斯 | 77.53% | 82.76% | 0.80 |
| 线性回归 | 82.74% | 79.63% | 0.81 |
| 支持向量机 | 83.68% | 82.44% | 0.83 |
| 决策树 | 95.47% | 94.89% | 0.95 |
| 随机森林 | 96.09% | 95.43% | 0.96 |

表3的实验结果表明，随机森林分类器可以取得最好的结果，分类准确率与召回率达到了95%。

学习曲线是描述机器学习模型效果随着参与训练的数据集的比例变化的曲线，从曲线中可以看出模型的训练准确率与交叉验证准确率，从这两项参数中可以看出模型在训练过程中是否发生了过拟合或欠拟合的情况。在CTU-13与ISCX-VPN数据集上使用随机森林分类器计算得出的学习曲线如图5所示



图表 5 随机森林分类器学习曲线

学习曲线表明，模型训练准确率与交叉验证准确率随着数据量的增多不断靠近，最后稳定在0.95-0.96之间，未出现过拟合现象。即使使用传统的机器学习分类器，基于时间窗口的特征提取方案也使得僵尸网络流量与正常的网络通信具有了良好的区分度。

## 基于TCPSession

TCP会话是一个TCP连接中所有数据包的有序集合，是计算机网络程序进行联网活动的基本单位，其内在的统计特性与传输层协议信息可以在一定程度上反映出程序的网络行为特征， 所以我们选择TCP会话为流量基本单位来完成流量识别任务。使用基于TCP会话的统计特征来描述僵尸网络行为比单向的TCP流更加准确，但是无法处理传输层协议之下的包，这部分数据包处理能力的缺失需要基于时间窗口的特征提取方案来补充。基于TCPSession的特征列表如表3所示

表格 4 TCP会话特征

|  |  |  |
| --- | --- | --- |
| 特征名称 | 含义 | 用途 |
| Duration | TCP会话持续的时间 | 协议特征、相似性检测 |
| FPS | 第一个数据包的大小 | 协议特征 |
| ATC | 数据流之间的平均间隔时间 | 相似性检测 |
| MPL | 数据流中最大的数据包大小 | 协议特征 |

Duration是TCP流持续的时间，有些软件使用短连接，建立TCP连接传输必要数据之后连接断开，比如HTTP短连接模式，有些软件使用长连接，程序会使用同一条TCP连接传输多块数据，比如心跳检测。通过流持续时间可以一定程度上区分协议特征。

这些特征与4.2节的特征组合起来，一起作为识别TCP流数据的特征依据。但是单个的流特征反映僵尸程序的行为依然有困难，结合4.3节基于时间窗口特征提取中的步进滑动窗口思想，我们选择某个流的上下文的多个流，一起作为判定僵尸网络流量的依据。选取合适的窗口大小，使用固定大小的队列维护该窗口的流特征数据。

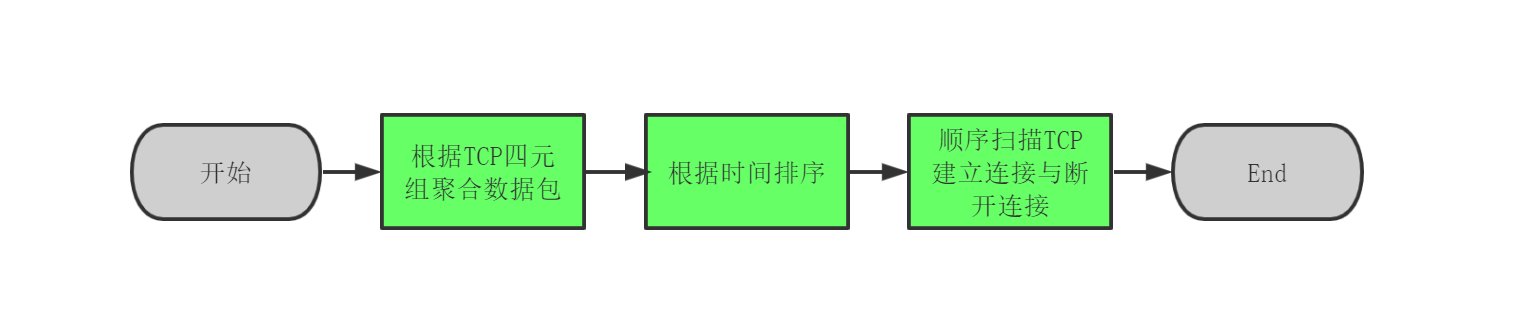
窗口大小的选择上，我们实现窗口大小与步进间隔可变的流特征提取工具，通过修改遍历窗口大小与步进间隔的参数组合，来找到合适的时间窗口和步进间隔。

### TCPSession提取方法

如何实时的从网络流量数据中提取TCPSession，是我们在特征提取之前需要解决的问题。Yang[37]等人在2019年提出了一种双向TCP流的提取方案，该方案兼顾了日常单线程TCP流提取与并行TCP流提取的需求，并且可以应用在大数据处理框架Spark中，为我们实时的处理大流量数据提供了重要参考。

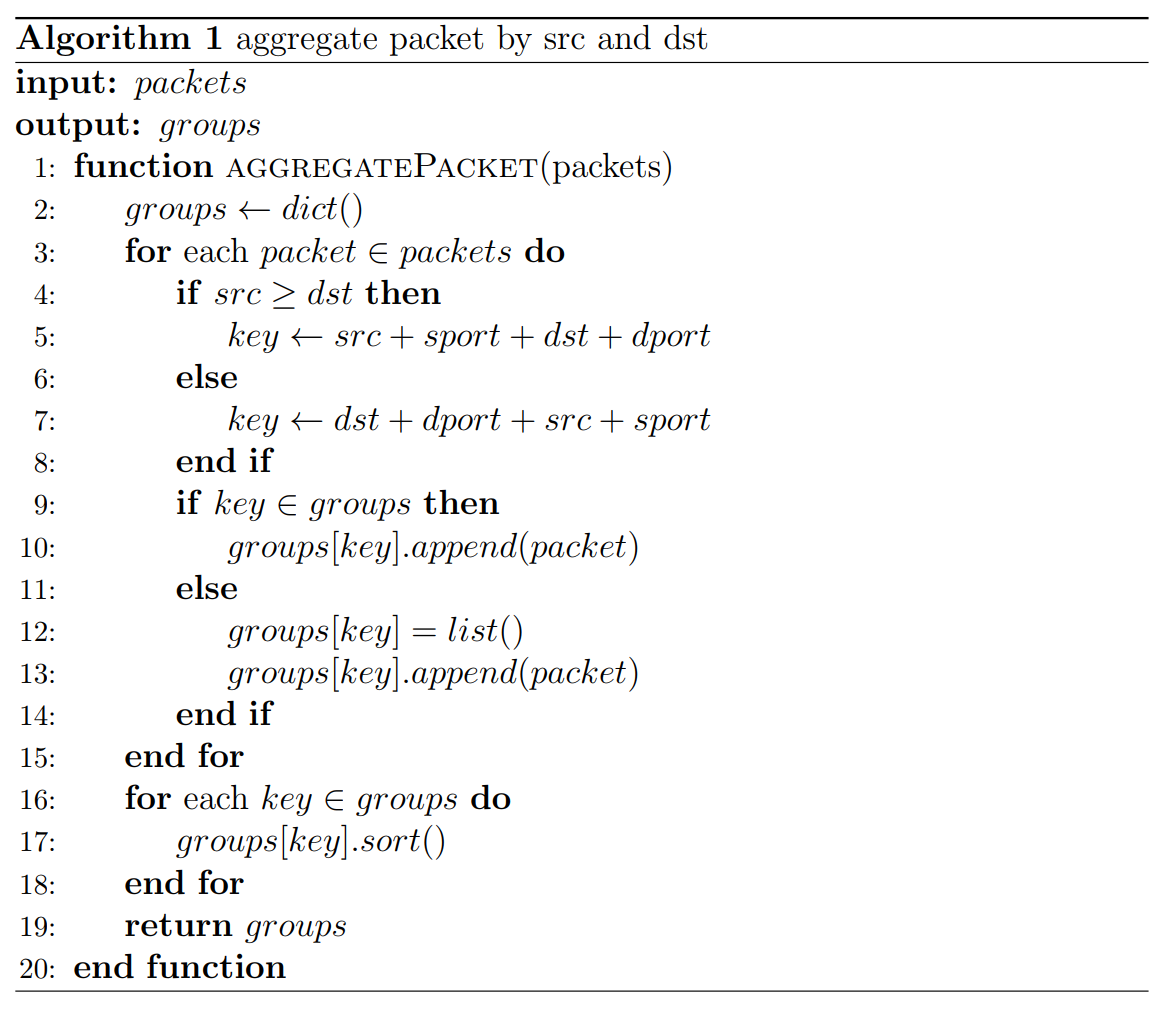
在单线程处理数据包时，可以采用基于TCP状态机的原理流式处理数据包，将每个数据包分配到从属的TCPSession之中，这种方法提取TCP流的方式较为准确，但是效率不高，无法使用分布式计算帮我们提高程序提取TCP流数据的效率，难以处理每秒数个T的流量数据。也可以基于TCP四元组将数据包进行分组，根据时间排序，再扫描一遍TCP连接建立数据包和TCP连接断开数据包，以此实现TCP流的提取，这种方法较为高效，也可以使用Spark的分布式处理能力，分布式提取TCP流数据。

使用该方案提取TCP流的关键步骤如图5所示



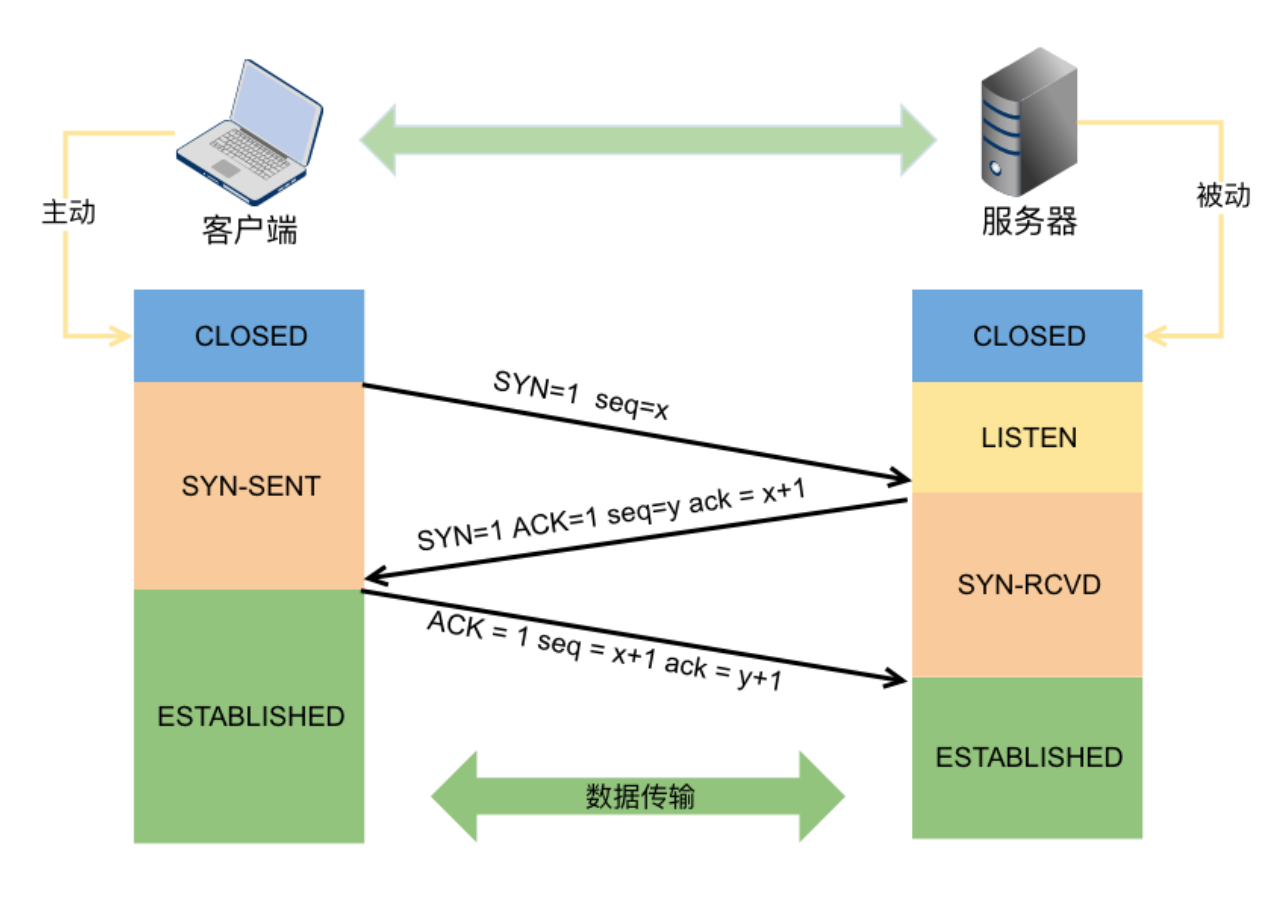
图表 6 提取TCP流关键步骤

将TCP数据包根据四元组聚合（src，sport，dst，dport）时需要注意的一点是TCP流是双向的，发送的数据包和接收的数据包的源地址和目的地址正好是相反的，我们需要准确的将数据包分组，就要完美的处理这种顺序的问题。Yang[37]等人提出的方案是将数据包的四元组信息提取出来，将src与dst根据字典序排序，得到该数据包从属数据流的key，使用该key作为索引，去找到我们已经聚合的数据包分组，将该数据包加入该分组中。聚合数据包的伪代码如图6所示



图表 7 聚合数据包的伪代码

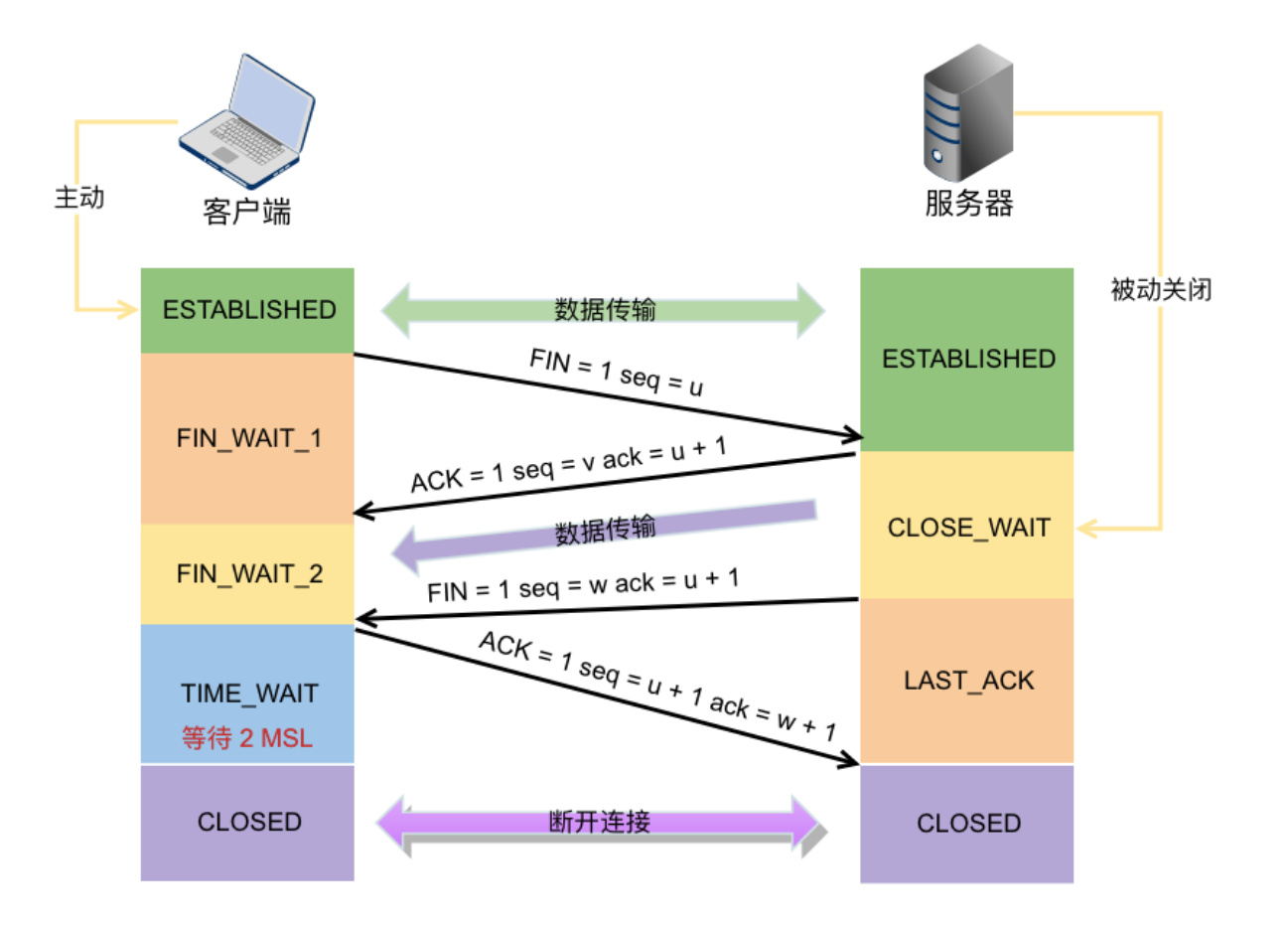
从数据包中提取到源地址、源端口、目的地址、目的端口，如果源地址大，那么源地址在前构成该session的key，如果目的地址较大，用目的地址在前构成该session的key，这样保证了一个数据流双向的数据包生成的key是相同的。将数据包分组之后，再将每个数据包根据时间排序，得到的每个分组都是若干个连续流的集合。此时我们需要从前往后扫描数据包，找到连接的建立与关闭步骤，从这些位置划分流的开始与结束，实现TCPSession的分割。TCP协议通过三次握手建立连接，其过程如图7所示



图表 8 TCP三次握手示意图

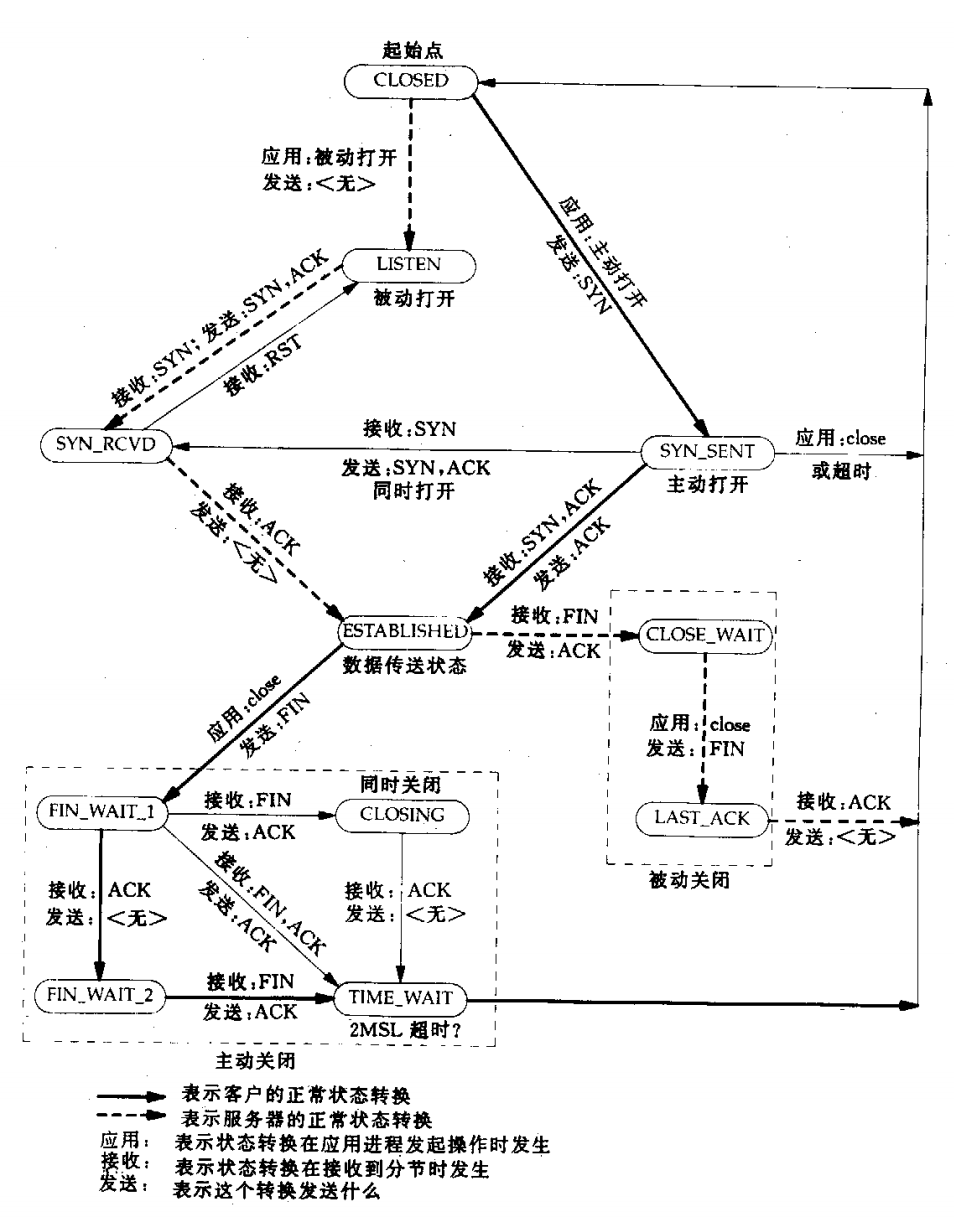
建立连接的三次握手过程中，第一二次握手确保接收端可以收到来自发送端的消息，第二三次握手确保发送端可以接收到接收端的消息。

TCP断开连接使用四次挥手，其过程如图8所示



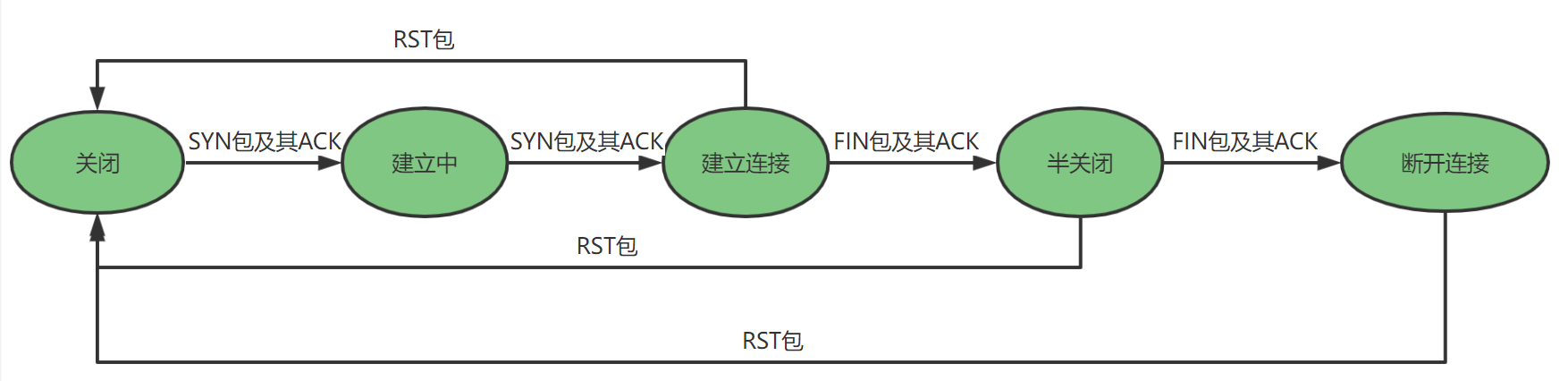
图表 9 TCP四次挥手示意图

四次挥手中，前两次关闭某个方向的写入流，后两次关闭另一个方向上的写入流，TCP连接关闭。通过提取TCP数据包标志位可以得知该数据包的类型，我们可以根据该数据包的标志位以及当前连接所处的状态来判断接收到该数据包时数据流状态的变化情况。完整的TCP状态转移如图9所示



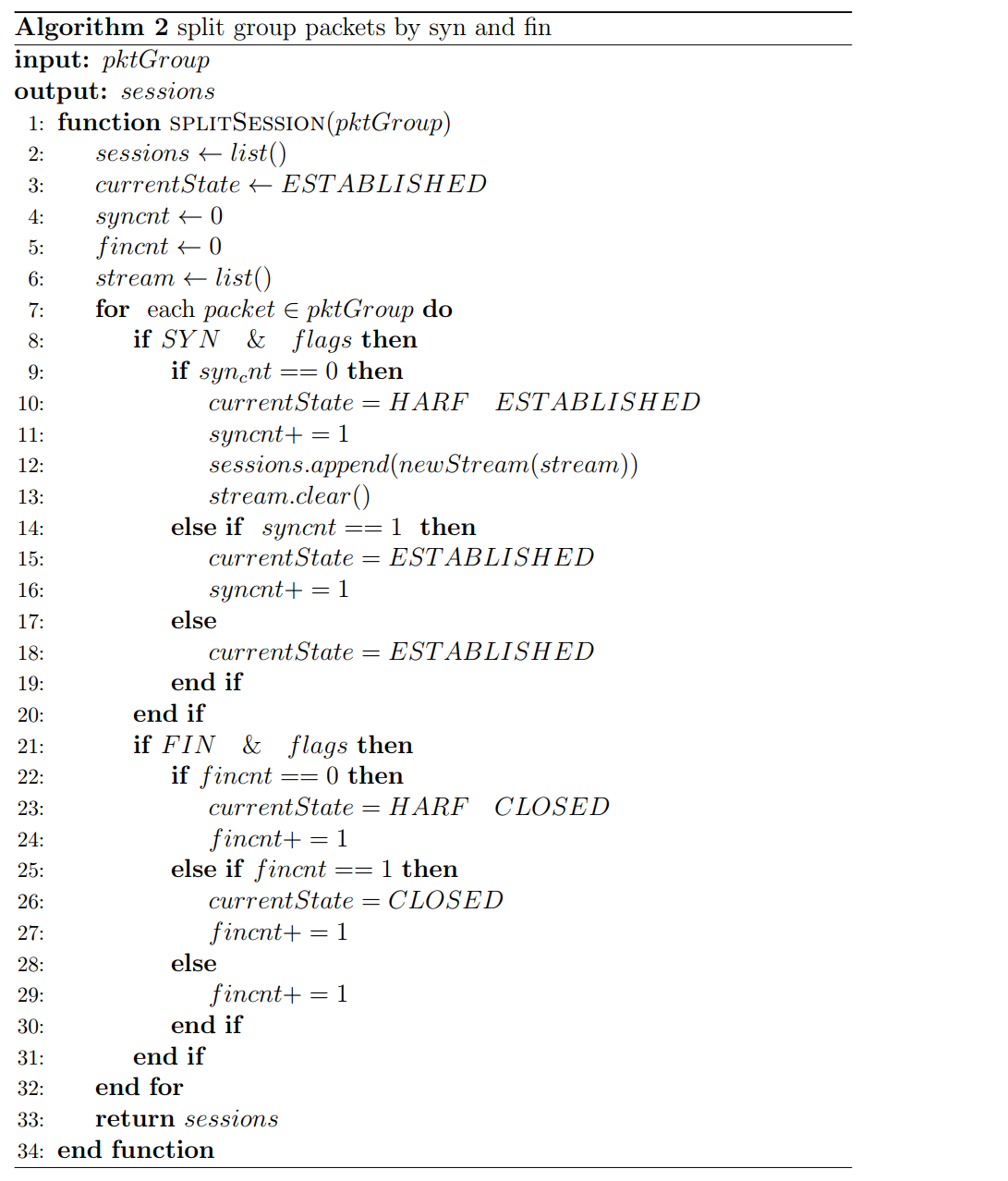
图表 10 TCP状态转移图

带有SYN标志的包及其ACK的出现表示一个TCP连接的开始，接下来的反方向的SYN包及其ACK包的出现表示TCP连接的建立。带有FIN标志的包及其ACK的出现表示一个TCP连接某个方向上的关闭，带有FIN标志的另一个包及其ACK的出现表示另一个方向上连接关闭。依据建立连接与断开连接的标志位，将首尾相接的流数据分隔开。该操作中，状态转移如图所示



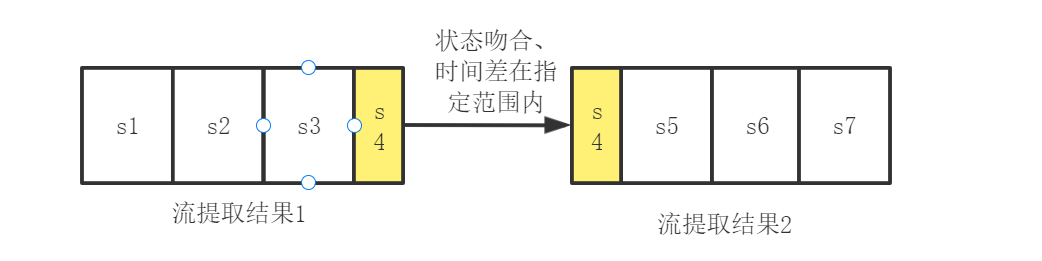
图表 11 分割TCP流的状态转移图

TCP流的四种状态被规定为连接建立中、已建立连接、断开中、已断开连接。建立连接中表示首组SYN的确认，已建立连接表示第二组SYN包的确认，断开中表示第一组FIN包的确认，已断开表示第二组FIN包的确认。根据状态转移图，从同一四元组的数据包集合中分割TCP流的伪代码如下



图表 12 分割TCPSession伪代码

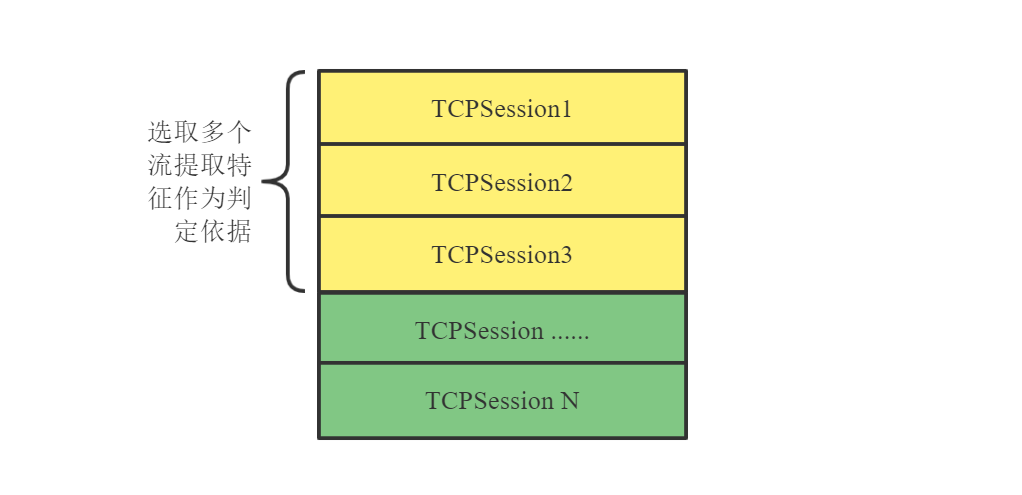
实时的流量处理需要用到大数据处理框架做分布式计算，否则难以应对大流量的处理需求，而分布式提取TCP流会面临一个问题，由于数据包文件被分割成了若干个小文件，有可能从属于同一个TCP流的数据包会被分配到不同的数据块中，这些数据块被单独的提取TCP流后会产生半个TCP流的情况，需要结合其他数据块的提取结果修正被分割开的TCP流。由于将数据分割成数据块的操作可以根据时间将原本的大数据包文件分成小文件，小数据包文件经过流提取操作得到的数据流集合，依然具有时间信息，每个小数据包文件流提取的结果在时间上首尾相连即可，处理这些小文件接缝处的TCP流（被分割开的单个流）时，注意前后流的状态吻合。



图表 13 单个被分割流的整合示意图

### TCPSession特征提取

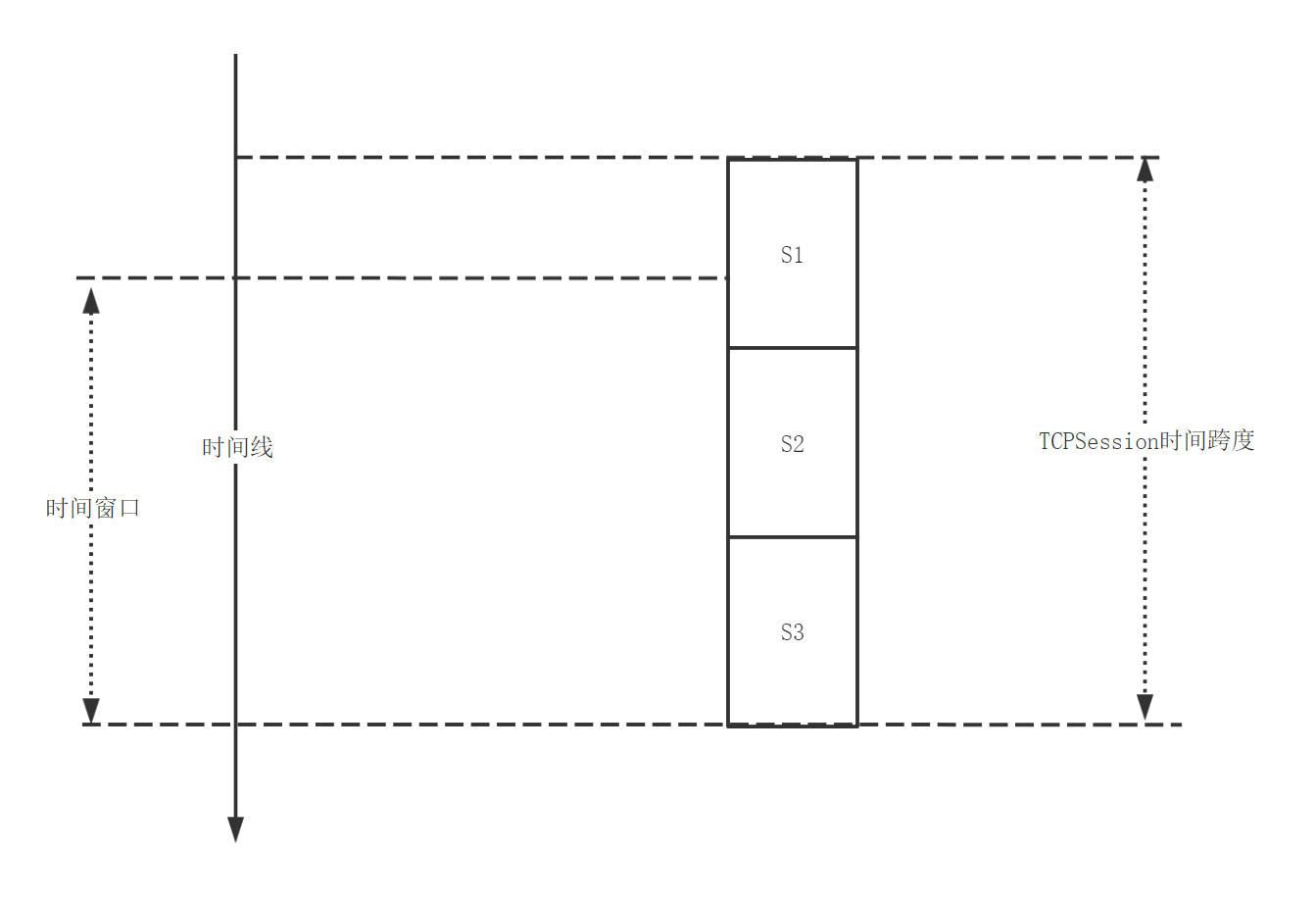
由于单个的TCPSession描述程序的网络行为依旧有困难，TCPSession的特征提取我们依旧采取TCPSession上下文的方式，将某个数据流与固定数量的从属该主机的其他数据流一起，作为判定该主机是否具有恶意行为的依据。



图表 14 TCPSession上下文示意图

处理前几个session时，不具有充足的上下文信息，只需要将Session上下文信息使用第一个session的特征补齐即可。比如上下文的宽度设定为3， 在处理第一个session时，将第一个session的内容填充三次，获得补全的上下文信息，再输入模型做预测。

使用TCPSession上下文作为特征提取单元时，其时间跨度可能会大于时间窗口，能为我们提供更多的历史信息，更加准确的描述应用程序的网络行为。

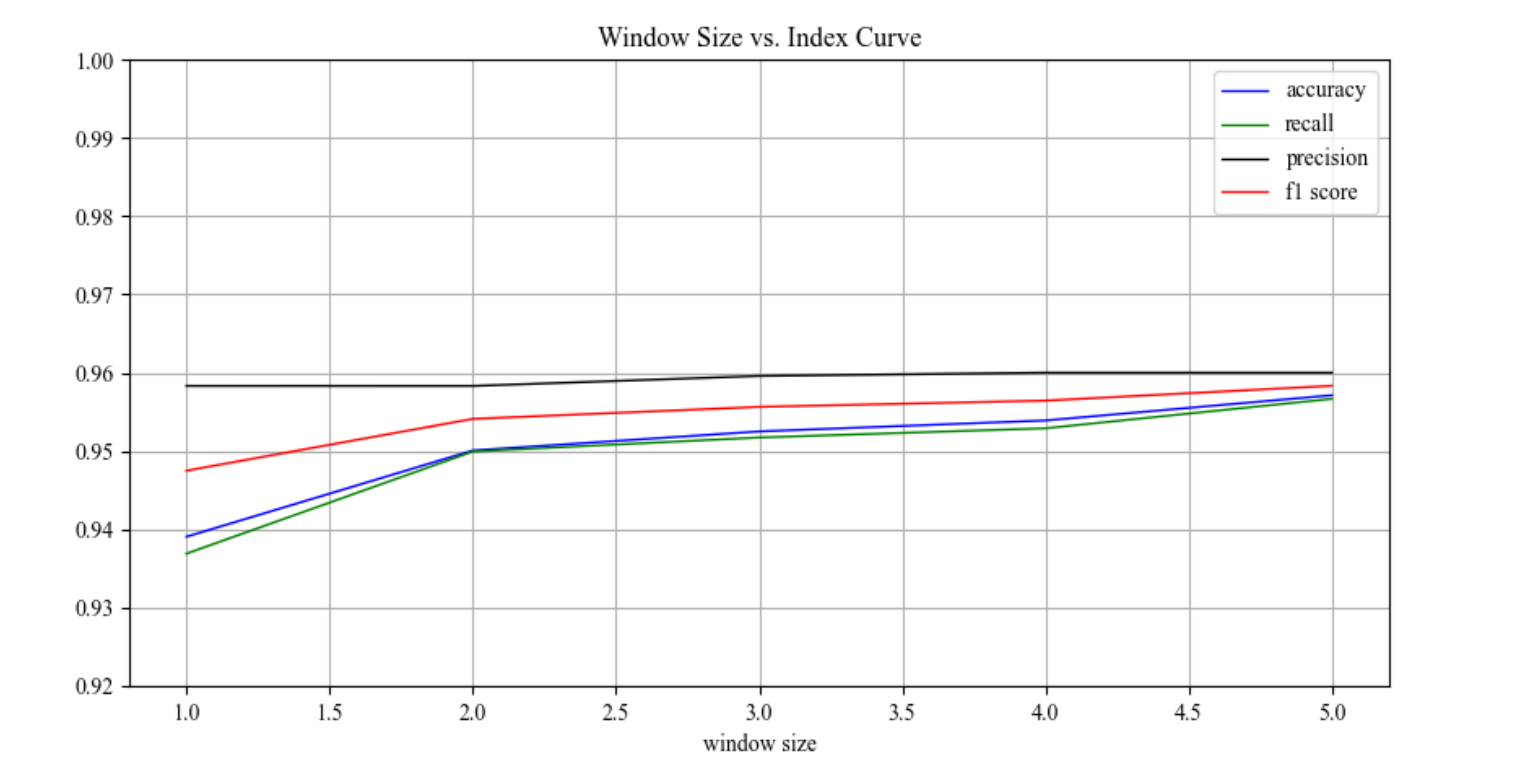


图表 15 时间跨度示意图

同样的，我们需要根据不同的上下文窗口大小提取出的特征在常规分类器上的表现选择合适的上下文窗口大小。

使用4.4.1节的TCPSession提取方式从CTU-13中提取僵尸网络TCP会话数据 12108条，作为恶意样本，从ISCX-VPN数据集提取TCP会话数据5253条，作为正常样本。可以看出我们提取到的恶意样本数量是正常样本数量的两倍以上，如果直接输入决策模型训练，分类结果会有所偏差，所以我们每次训练模型时，随机从恶意样本中提取5253条样本数据，训练模型，重复若干次，训练N个机器学习模型，在预测时，从这N个模型的预测结果中选举出最终的预测结果。

我们使用随机森林分类器，绘制查准率、召回率、准确率、f1 score随着上下文窗口大小的变化曲线，如图16所示



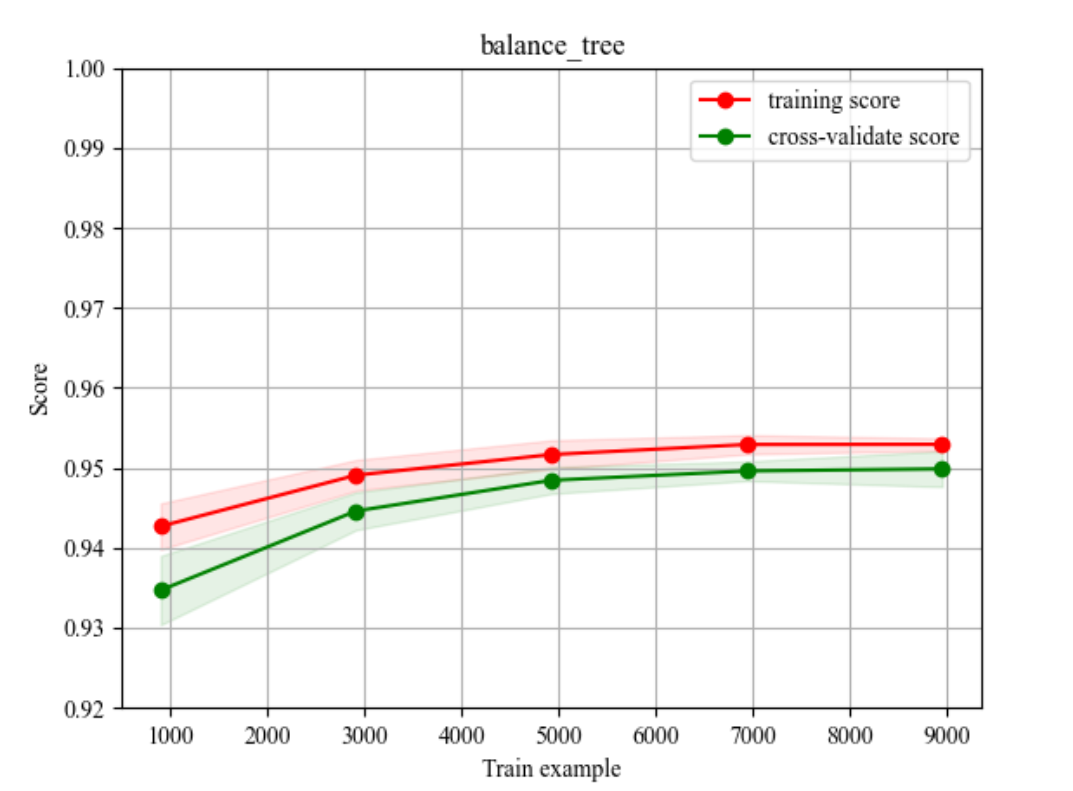
图表 16 分类结果与上下文窗口大小变化曲线

查准率、召回率与f1 score随着上下文窗口大小都有提升，但是在TCPSession上下文窗口大小达到3以后，准确率的上升十分有限，上下文窗口的大小已经不是影响分类效果的主要因素。因此我们选择3作为TCPSession上下文窗口大小的参考值。随后，使用多种传统分类器测试分类效果，得到的分类结果如表5所示

表格 5 传统分类器在TCPSession上下文上的分类结果

|  |  |  |  |
| --- | --- | --- | --- |
| 分类器 | 查准率 | 召回率 | f1 Score |
| 贝叶斯 | 89.57% | 86.35% | 0.88 |
| 线性回归 | 76.85% | 79.74% | 0.78 |
| 支持向量机 | 83.62% | 81.93% | 0.83 |
| 决策树 | 94.38% | 92.66% | 0.94 |
| 随机森林 | 95.05% | 95.33% | 0.95 |

使用选举模式的随机森林分类器在TCP会话上的分类效果更好，准确率与召回率达到了95%，同样的我们绘制该分类器的学习曲线如图17所示



图表 17 平衡树分类器学习曲线

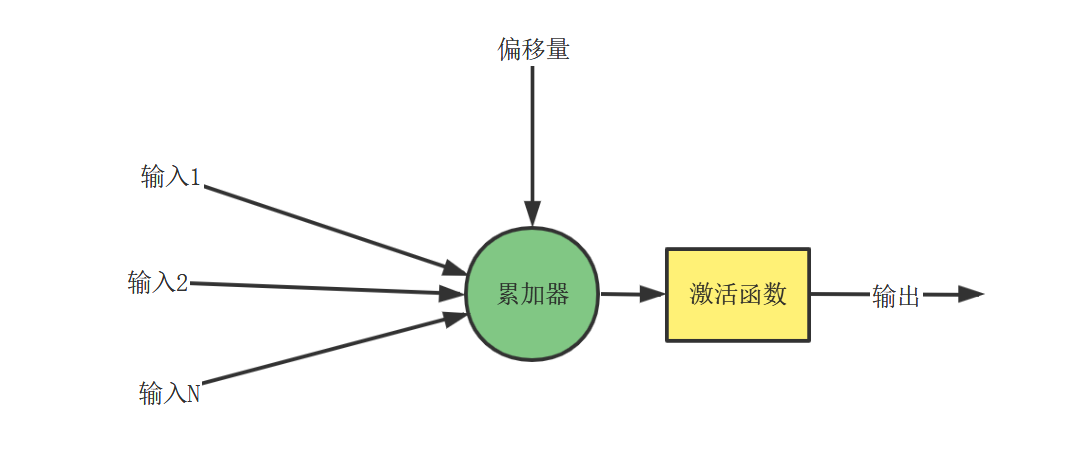
从学习曲线可以看出，随着样本数目不断增加，模型准确率缓慢升高至95%，且训练样本准确率与交叉验证样本准确率相近，模型未出现过拟合与欠拟合的现象。使用TCPSession的特征提取方式，使用上下文窗口内的会话训练模型，即使的传统的及其学习模型上也可以达到良好的区分僵尸网络流量的效果。

# 基于时间窗口与TCPSession的复合分类器

## 循环神经网络

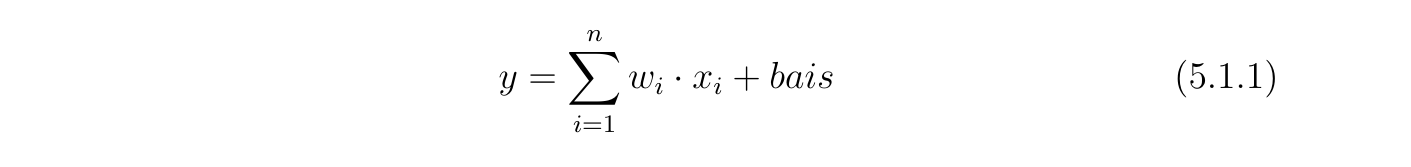
在第四节中，我们使用传统的机器学习方法对我们的特征选择方案进行了测试，结果表明基于时间窗口与TCPSession的特征选择方法从僵尸网络流量中提取到的特征，使用关联上下文的方式结合上下文窗口的多个特征后，与正常网络流量的特征具有较为明显的区分度，分类准确率达到了95%。但是上述方案仍然具有不足之处。僵尸网络的行为各异，每种行为产生的网络流量的时间跨度不同，TCP会话的个数也不尽相同，我们使用了确定的上下文窗口大小中的数据特征去刻画时间跨度不同的僵尸程序的网络行为，得到的结果也会具有一些局限性。那有没有更好的方式去解决这个上下文关联的问题呢？答案是循环神经网络RNN。

为了阐明循环神经网络，我们需要从神经网络开始说起。人工智能技术被设计用来使程序具有像人脑一样的智能行为，其主要载体就是机器学习算法与神经网络模型。在计算机程序中可以做出类似人脑智能决策的结构，叫做神经网络。并且我们将组成神经网络的基本单元形象化的称呼为神经元。其图像化表述如图18所示



图表 18 神经元结构示意图

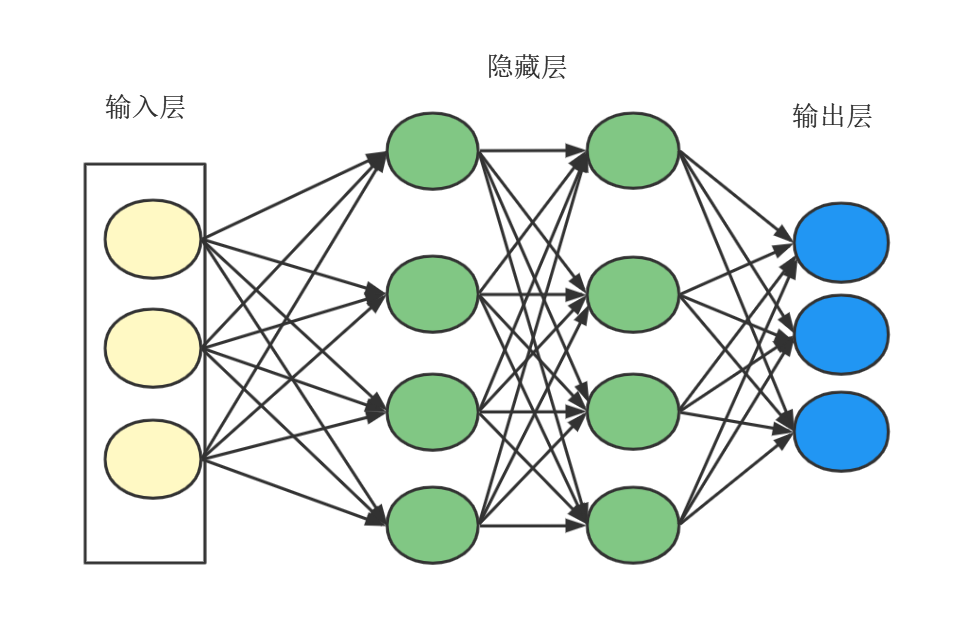
每个神经元有若干个输入，每个输入都对应一个权重，有一个修正神经元的偏移量，以上这些信号经过累加器之后输入激活函数，如果符合了激活函数的激活条件，该神经元会将输入的信号输出出去，如果未满足激活条件，则不会输出。累加器输出的结果如公式5.1.1所示



输入的向量与权重矩阵相乘后，加上修正偏移量，得到的值会输入到激活函数中。引入激活函数的目的是为了给神经元添加非线性因素，解决线性模型表达能力不足的缺点。最简单的激活函数是ReLU，当输入小于0时，激活函数不会被激活，输入大于0时，输出与输入相同。累加器实现输入数据与权重矩阵的线性运算，激活函数为神经元引入非线性因素，增加模型的表达能力。

神经网络就是由神经元前后相连组成的规则的网状结构，根据神经元类型与神经网络结构与功能的不同，神经网络又分为全连接神经网络、卷积神经网络与循环神经网络。其中，全连接神经网络是神经网络的原始结构，卷积神经网络与循环神经网络都是全连接神经网络针对特定领域问题的特化。全连接神经网络用于通用的分类与预测任务，卷积神经网络被专门用来解决图像类问题，循环神经网络用于处理前后数据有所关联的任务。

全连接神经网络是最简单的一类神经网络，由一层一层的神经元组成，下层的每一个神经元都保留了到上一层所有神经元的连接，所以叫全连接神经网络，其结构如图19所示

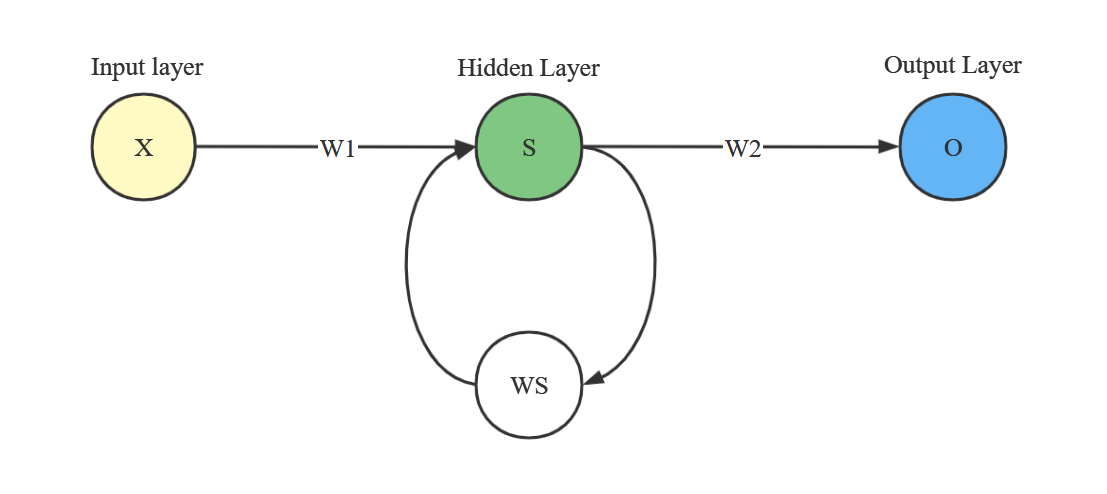


图表 19 全连接神经网络结构示意图

全连接神经网络包含输入层、隐藏层和输出层。输入层负责读取我们输入到神经网络的数据，其大小对应我们输入数据的维度。隐藏层会有若干层，每层都有数量相同或不同的神经元，每个神经元与上层的任意一个神经元保持连接。输出层负责处理隐藏层的输入，并且输出神经网络的识别结果。每个隐藏层神经元与输出层神经元都有自己的权重矩阵与偏移量。权重矩阵与偏移量是神经元的主要参数，我们在训练神经网络时，要寻找的主要内容，就是隐藏层神经元和输出层神经元的相关参数。全连接神经网络是最简单的一种神经网络，只要输入合适的数据，该类型的神经网络可以和机器学习算法一样完成预测与分类任务。可以将全连接神经网络当做一个可以拟合任何函数的黑盒，只要给足训练数据，神经网络就可以根据给定的输入输出我们预期的结果。

全连接神经网络结构简单，在充足的训练数据上有非常好的拟合效果，但该类型的神经网络不具有记忆数据的能力，是无状态的。全连接神经网络只能处理那些前后数据没有关系的任务，对于多条数据之间有某种联系的数据，该类型的神经网络的拟合效果不尽人意。比如我们在识别语义的时候，单独分析某个词的意思我们无法识别出整句话的含义，只有我们联结所有单词的含义，才能识别出整句话的意思。为了能更好的处理这种前后相关联的数据，RNN（Recurrent Neural Network）诞生了。

与全连接神经网络不同的是，循环神经网络的输出不仅与当前输入的数据有关，还与当前神经网络的状态有关。循环神经网络根据输入的数据和当前自身的状态信息给出对应的输出，并且更新自身的状态。也正是因为了循环神经网络有当前自身状态这一属性，所以该网络具有记忆之前数据的能力，处理时间上关联的数据时，其效果要优于全连接神经网络。其结构如图所示



图表 20 循环神经网络结构示意图

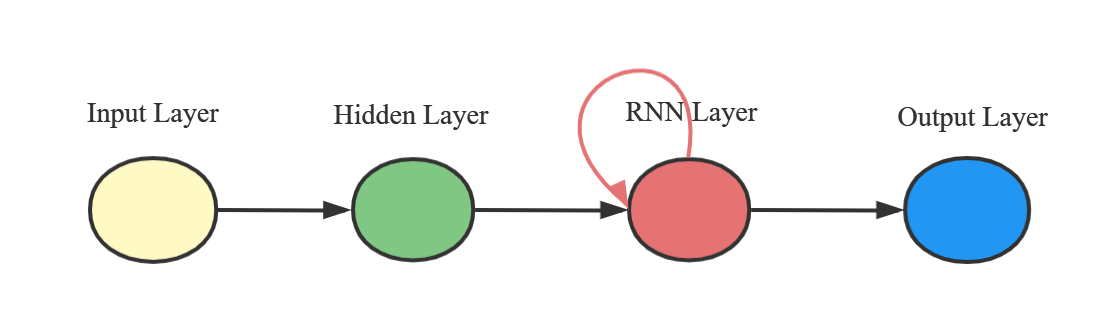
X代表输入向量，W1代表输入层到隐藏层之间的权重矩阵，S表示隐藏层当前状态，W2表示隐藏层到输出层的权重矩阵，O表示输出层的输出。WS也是一个权重矩阵，该神经网络此时此刻的输出，由该神经网络上一时刻的状态St-1与WS权重矩阵的乘积，与当前时刻该神经网络的输入Xt共同决定。所以，循环神经网络当前时刻的输出与先前的输入的历史数据有关，且历史数据对当前输出的影响具体体现在权重矩阵WS。

循环神经网络可以关联先前输入数据这一点，与我们识别僵尸网络行为模式的需求十分吻合。僵尸网络的网络行为具有一定的时间跨度，一段时间前后的数据包或数据流具有关联性，使用循环神经网络理论上可以更好的实现僵尸网络流量的异常检测。

## 检测模型

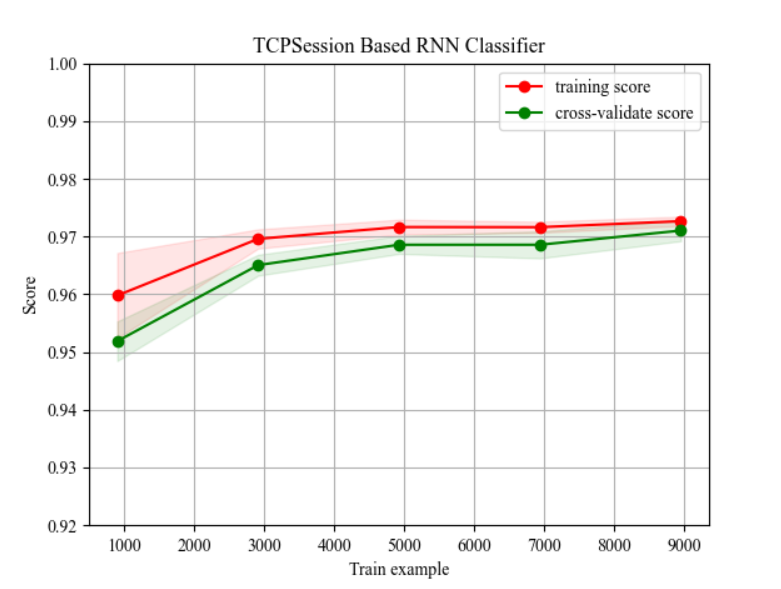
在4.4节中，我们使用TCPSession上下文的方式，让当前的TCPSession关联部分历史数据，实现僵尸网络流量的检测，但是我们的上下文窗口是固定值，该选择多少TCPSession来作为判定僵尸网络流量的数据输入，是我们在CTU-13数据集与ISCX-VPN数据集上计算得出的固定值，但是僵尸网络行为多样，不同类型的动作会产生数量、内容不同的数据流，通过这种固定上下文窗口的方式检测僵尸网络流量具有一定的局限性。循环神经网络是神经网络的一种，属于有状态的神经网络，具有记忆历史数据的功能，神经网络对当前数据的处理结果还被先前输入过的数据所影响，与我们通过僵尸网络流量上下文识别僵尸网络流量的需求相吻合，所以我们选择循环神经网络作为我们TCPSession的分类器，对于基于时间窗口的检测方案，因为时间窗口数据包的统计特征已经包含了上下文信息，所以时间窗口上的检测方案依旧采用随机森林的分类器。

基于RNN构建TCPSession分类器，输入层具有20个神经元，对应我们输入数据的特征数目，接下来是一层64个单元的全连接层，后面接含有64个单元的RNN层，然后是含有两个单元的输出层，其结构构成如图21所示



图表 21 TCPSession RNN分类器结构图

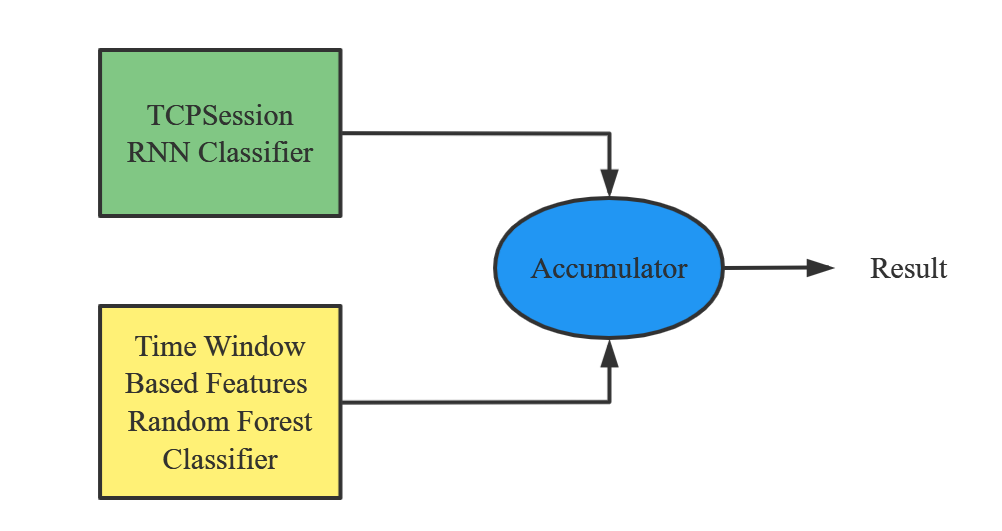
由于TCPSession的恶意样本与正常样本不均衡，我们仍然采用多个分类器，每次随机抽取数量相同的样本训练模型，得到若干个模型，最终使用若干个模型的选举结果来表示最终的判定结果。基于TCPSession的RNN分类器在该数据集上的学习曲线如下



图表 22 RNN分类器学习曲线

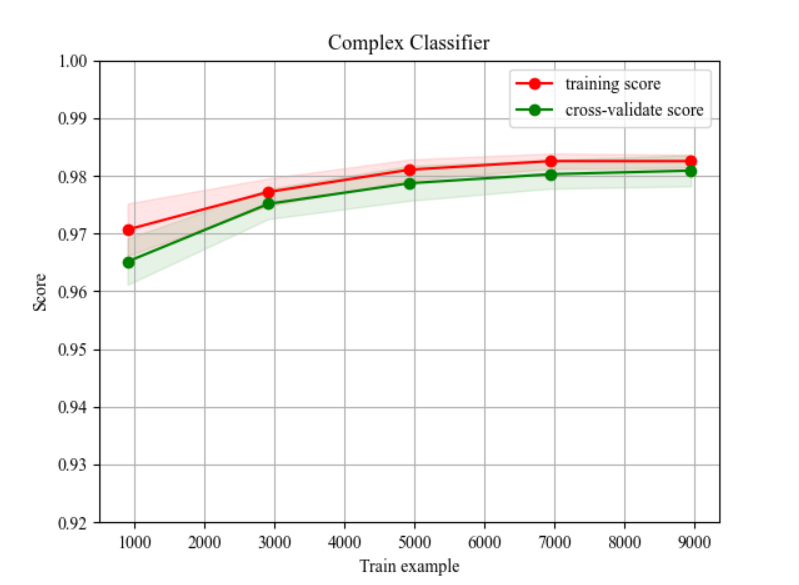
实验表明，该分类器在CTU-13与ISCX-VPN数据集上提取到的特征数据上分类准确率达到了97%，在该数据集上的区分度优于随机森林分类器。

虽然基于TCPSession的RNN分类器区分僵尸网络流量与正常网络流量上具有成效，不过对传输层协议之下的数据包无效，对UDP数据包的分析也被忽略，针对这些数据包的分析需要基于时间窗口的分类器来补充。可以组合两个分类器的分类功能，实现僵尸网络流量检测。我们记录两个分类器针对同一个IP地址的判定结果，如果两个分类器都认为该主机具有僵尸程序的行为，那么该主机可疑等级为2， 如果仅有一个分类器认为该主机具有僵尸程序行为，那么该主机可疑等级为1，如果两个分类器输出都表示未检测到该主机有可疑行为，那么该主机可疑等级为0，表示正常主机。



图表 23 复合分类器结构示意图

将两个分类器的检测结果组合起来，如果将任一模型判定的恶意流量都认为是僵尸网络流量，可以提高我们模型的召回率，找到更多的僵尸网络流量，模型表现的比单个分类器更加严格，不过也付出了误报率略微提高的代价。复合分类器在CTU-13与ISCX-VPN数据集上的学习曲线如图所示



图表 24 复合分类器学习曲线

在我们的实验中，使用如图22所示的复合分类器在CTU-13与ISCX-VPN数据集上分类准确率达到了98.4%，召回率达到了98%，总体效果优于使用单个RNN分类器的情形。召回率更高意味着从网络流量中识别僵尸网络流量的能力更强，检测效果更好。模型输出主机的可疑级别信息，对于可疑级别处于1，仅有一个模型判定为僵尸网络流量的主机，我们可以结合其他维度的信息给出更准确的判断结果。比如通过检测增值网络攻击的方式，判断该主机是否具有其他恶意行为。

## 模型评估

僵尸网络检测系统的首要任务，是能检测出僵尸主机。与检测出僵尸主机的速度相比起来，能更加准确地输出可疑僵尸主机的信息才是最重要的。僵尸程序也是计算机程序，它们有些行为是会和正常的网络程序相重叠，甚至有的僵尸程序会在自己的活动中随机加入正常网络行为来躲避僵尸网络检测系统，比如随机的访问公共网络，随机的发送多种类型的网络流量。当僵尸网络产生这种行为的时候，我们的僵尸网络检测系统会产生误判也是正常的结果，当僵尸程序进行恶意网络活动或连接C&C控制服务器的时候僵尸网络检测系统可以检测出来，就足以说明该检测系统的检测能力。所以，与识别每个单位的TCPSession与时间窗口特征的准确性相比，笔者更看重该模型是否可以正确检测出僵尸主机的能力。

我们使用CTU-13与ISCX-VPN数据集训练出的模型在这两个数据集中分离出的测试集分类准确率达到了98%，但是CTU-13数据集虽然包含了众多僵尸程序的活动场景，但是该数据集包含的僵尸程序类型有限，在该数据集上检测效果并不能说明该模型在实际场景下的检测效果如何。ISCX-Botnet数据集是纽布伦斯威克大学网络安全研究所混合多种僵尸网络流量与正常的网络背景流量创建的数据集，与注重僵尸程序活动场景不同，该数据更在意数据集中包含僵尸程序的数量与僵尸网络流量与正常网络流量的组合，力求模拟真实的网络环境的流量状况，该数据集的训练集包含7中不同的僵尸网络流量，测试集包含16种不同的僵尸网络流量，恶意样本丰富，混合了正常的互联网背景流量后，符合僵尸网络流量在互联网上出现的真实场景，是相比于CTU-13更加完备的数据集，我们使用该数据集衡量我们模型的检测效果。该数据集中包含的僵尸网络流量类型数据如表格所示

表格 6 ISCX-Botnet训练集包含数据流统计信息

|  |  |  |  |
| --- | --- | --- | --- |
| **僵尸网络名称** | **僵尸网络类型** | **数量（条）** | **占比（%）** |
| Neris | IRC | 21159 | 12 |
| Rbot | IRC | 39316 | 22 |
| Virut | HTTP | 1638 | 0.94 |
| NSIS | P2P | 4336 | 2.48 |
| SMTP Spam | P2P | 11296 | 6.48 |
| Zeus | P2P | 31 | 0.01 |
| Zeus（C&C） | P2P | 20 | 0.01 |

表格 7 ISCX-Botnet测试集数据流统计信息

|  |  |  |  |
| --- | --- | --- | --- |
| **僵尸网络名称** | **僵尸网络类型** | **数量（条）** | **占比（%）** |
| Neris | IRC | 25967 | 5.67 |
| Rbot | IRC | 83 | 0.018 |
| Menti | IRC | 2878 | 0.62 |
| Sogou | HTTP | 89 | 0.019 |
| Murlo | IRC | 4881 | 1.06 |
| Virut | HTTP | 58576 | 12.8 |
| NSIS | P2P | 757 | 0.165 |
| Zeus | P2P | 502 | 0.109 |
| SMTP Spam | P2P | 21633 | 4.72 |
| UDP Storm | P2P | 44062 | 9.63 |
| Tbot | IRC | 1296 | 0.283 |
| Zero Access | P2P | 1011 | 0.221 |
| Weasel | P2P | 42313 | 9.25 |
| Smoke bot | P2P | 78 | 0.017 |
| Zeus(C&C) | P2P | 31 | 0.006 |
| ISCX IRC Bot | P2P | 1816 | 0.387 |

从图表中可以看出，ISCX-Botnet数据集中包含了大量的IRC、P2P僵尸网络流量与少部分的HTTP僵尸网络流量，并且测试集中包含的僵尸网络类型更多，可以用来检测模型识别未曾见过的僵尸网络的能力。

我们将ISCX-Botnet数据集处理成基于时间窗口的特征与基于TCPSession的特征。由于ISCX-Botnet数据集的时间跨度较长，在一些时间段中的数据包数量很少，在5s的时间间隔内只有个位数的数据包或没有数据包，对提取统计特征贡献不大，我们略去这部分数据，在时间窗口设置为10，步进间隔设置为5，提取基于时间窗口的统计特征。在良性流量与恶意流量的区分上，我们将所有发往恶意主机与来自恶意主机的流量都判定为恶意流量。最终得到流量特征的统计数据如表格所示

表格 8 ISCX-Botnet训练集提取特征数量表

|  |  |  |
| --- | --- | --- |
| 特征类型 | 僵尸网络流量（条） | 背景流量（条） |
| 时间窗口 | 189745 | 276580 |
| TCPSession | 51589 | 139847 |

表格 9 ISCX-Botnet测试集提取特征数量表

|  |  |  |
| --- | --- | --- |
| 特征类型 | 僵尸网络流量（条） | 背景流量（条） |
| 时间窗口 | 2134037 | 7647581 |
| TCPSession | 21959 | 122102 |

我们将提取到的特征数据输入5.2节阐述的检测模型中，在ISCX-Botnet 训练集与测试集上得到的检测结果分别如表格所示

表格 10 ISCX-Botnet训练集流量特征检测准确率

|  |  |  |
| --- | --- | --- |
| 分类器 | 准确率（%） | 召回率（%） |
| 复合分类器 | 87.49% | 85.46% |

表格 11 ISCX-Botnet测试集流量特征检测准确率

|  |  |  |
| --- | --- | --- |
| 分类器 | 准确率（%） | 召回率（%） |
| 复合分类器 | 79.83% | 78.62% |

根据图表中的实验结果，本文5.2节描述的复合分类器在ISCX-Botnet训练集上达到了87.5%的准确率，在测试集上达到了将近80%的准确率。虽然在ISCX-Botnet数据集上提取到的特征上的准确率不太乐观，但是笔者认为，相比较于识别出僵尸主机的每个网络行为产生的网络流量，僵尸网络检测系统重要的事能够识别出僵尸主机。在ISCX-Botnet数据集中共含有35个僵尸主机，其各种类型僵尸主机IP信息与复合模型检测效果如表所示

表格 12 不同类型僵尸网络IP地址信息及其检测结果

|  |  |  |
| --- | --- | --- |
| **僵尸网络名称** | **IP地址列表** | **检测出的IP地址** |

|  |  |  |
| --- | --- | --- |
| IRC（IRC） | 192.168.2.112  192.168.5.122 192.168.2.110 192.168.4.118 192.168.2.113 192.168.1.103 192.168.4.120 192.168.1.105 10.0.2.15 | 192.168.2.112  192.168.5.122 192.168.2.110 192.168.4.118 192.168.2.113 192.168.1.103 192.168.4.120 192.168.1.105 10.0.2.15 |
| Neris | 147.32.84.180 | 147.32.84.180 |
| Rbot | 147.32.84.170 | 147.32.84.170 |
| Menti | 147.32.84.150 | 147.32.84.150 |
| Sogou | 147.32.84.140 | 147.32.84.140 |
| Murlo | 147.32.84.130 | 147.32.84.130 |
| Virut | 147.32.84.160 | 147.32.84.160 |
| Black Hole | 10.0.2.15 192.168.106.141  192.168.106.131 |  |
| Tbot | 172.16.253.130 172.16.253.131 172.16.253.129 172.16.253.240 | 172.16.253.130 172.16.253.129 |
| Weasel | 74.78.117.238 158.65.110.24 | 158.65.110.24 |
| Zeus | 192.168.3.35 192.168.3.25 192.168.3.65 172.29.0.116 | 192.168.3.65 |
| Osx\_trojan | 172.29.0.109 |  |
| Zero access | 172.16.253.132 192.168.248.165 |  |
| Smoke bot | 10.37.130.4 | 10.37.130.4 |

从表格中的IP地址检测结果可以看出，我们的复合检测模型可以检测出所有的Neris、Rbot、Menti、Sogou、Murlo、SMTP Spam、UDP Storm、ISCX IRC Bot僵尸网络主机，部分Weasel、Zeus、Virut僵尸网络主机，对使用P2P协议的僵尸主机，该模型的检测能力较差。从IP地址检测效果来看，5.2节描述的复合模型，在训练集只有CTU-13数据集中的4中僵尸网络程序的活动场景前提下，可以检测出具有同类行为的僵尸网络主机，即使一些僵尸网络数据未在训练集出现，但是由于僵尸程序都具有的一些行为，使其暴露了自己在互联网上的行踪，比如Murlo僵尸网络。僵尸程序形态各异，在各种各样的状况下会产生多种行为，多种形态各异的流量数据，但是其基本模式是有迹可循的。

基于TCPSession的RNN分类器与基于时间窗口的随机森林分类器构成的复合分类器，具有学习僵尸网络行为的能力，即使未知的僵尸程序产生的流量数据尚未在训练集中出现过，也会因为其相同的行为模式在进行网络通信的某个步骤中被该模型检测出来。该模型虽然实时的检测僵尸网络流量数据的准确率有待提高，但是通过RNN学习僵尸网络行为模式的方法取得了一定效果，可以识别出具有恶意行为的僵尸主机。

虽然该模型具有了一定学习僵尸网络行为的能力，但是该模型也有一些自己的弊端。由于是有监督学习，尚未在训练数据集中出现过的僵尸网络，检测效果未知。如果新的僵尸程序具有和训练集中僵尸程序相似的行为模式，那么该模型可以检测出僵尸主机，但是如果僵尸程序具有新的行为模式，该模型不具有检测该种类型僵尸程序的能力。由于模型的检测能力来源是网络流量的统计特征，当僵尸程序在自己的通信数据中混入了自身大量的正常网络流量之中，那么僵尸程序的行为特征会被淹没在网络背景流量之中从而使在这些流量上的统计特征与正常网络流量的统计特征区分度很低，从而使得检测难以生效。除此之外，机器学习算法与神经网络运用在网络安全领域有一个天生的弊端，不可解释性，僵尸网络检测领域也不例外。某个流量结果的判定为恶意的，说明产生该流量的主机有可能是恶意的僵尸网络主机，但是我们无从得知这样判定的原因，僵尸程序的种类是什么，它到底是在僵尸程序活动的哪个阶段，在做什么事的时候被检测到了；某个区间的流量产生了漏报，我们无法确定漏报是如何产生的，同样也无法确定误报是怎样产生的，类似这样更加详细的有关该恶意软件的信息，需要我们使用更加多维的检测模型，结合其他更多维度的信息，综合的给出更加准确的判定。

# 总结与展望

僵尸网络是一群被攻击者控制的主机组成的网络，可以接收控制者的攻击指令，发起各种攻击活动，窃取重要敏感信息或传播自身，造成互联网的重大损失。僵尸网络发展至今，已经随着互联网安全技术的发展从最初单纯使用IRC协议具有中心化结构的僵尸网络，变成使用P2P和HTTP协议加代理技术的分布式僵尸网络，在演化成使用混合协议的分布式与中心化共存的混合结构僵尸网络，其健壮性获得了显著提高，为我们检测僵尸网络数据造成了阻碍。僵尸程序的更新也是伴随着防御技术的进步的，随意在僵尸网络的检测与防御领域，一直也是僵尸网络防御技术与恶意攻击者相互竞争的状态。

本文我们主要介绍了僵尸网络的运行机理，随着僵尸网络防御技术的发展而出现的P2P、HTTP类型僵尸网络的运行机制，僵尸网络常用的隐藏中心节点的手段。随后我们介绍了近二十年间经典的僵尸网络检测方案，并从中学习检测僵尸网络的基本方法思路。从僵尸网络检测方案中，笔者学习到，机器学习与神经网络是我们开展流量检测任务的重要手段，也是我们检测僵尸网络流量的重要工具。使用机器学习与神经网络的第一个重要步骤，就是特征的提取，如何提取特征甚至比如何构建检测模型更加重要，合适的特征提取方案可以表现出僵尸程序的行为特征，使得恶意僵尸网络流量数据相较于正常的网络背景流量数据具有更好的区分度，从而使得分类器具有更好的分类效果。所以，本文第三章提出了被多位研究者使用过的基于时间窗口统计特征的特征提取方案与基于TCP会话的特征提取方案，基于TCP会话的特征用于描述网络程序的网络行为，时间窗口上的特征用于补充TCP会话特征在UDP协议与传输层协议之下的流量上特征的缺失，笔者结合常见的机器学习分类器检测了这两种特征提取方案在分类CTU-13与ISCX-VPN数据集上的表现，实验表明这两种特征提取方案在僵尸网络流量数据集与正常的网络背景流量上的区分度达到了95%。随后我们基于这两种特征提取方法，构建分类器，基于时间窗口的特征提取方案我们选择了随机森林分类器，基于TCP会话的特征提取方案我们构建了可以处理时间相关数据的RNN分类器，同样是在CTU-13与ISCX-VPN数据集上，这两个分类器单独的分类准确率达到了97%，僵尸网络流量召回率达到96%。随后笔者组合了两个分类器的检测结果，使得模型具有更高的僵尸网络流量召回率，达到98%，增加了模型从背景网络流量中识别僵尸网络流量的能力。在构建了最终的模型之后，笔者在ISCX-Botnet数据集上进行了测试，虽然基于流量特征的数据准确率不高，只有80%，但是该检测模型检测僵尸主机的能力得到了确认。随机森林模型与RNN模型组成的复合模型，具有学习僵尸程序网络行为的能力，即使对一些尚未在训练集中出现过的流量数据，只要恶意程序与训练集中的僵尸程序具有了一定程度上相同的行为模式，该模型就可以检测出该主机。也就是说，该模型检测出了僵尸主机，但是并没有检测出该主机发出的所有流量。在评估模型的最后，我们也给出了该模型存在的一些问题，难以识别具有未知行为的未知恶意程序、结果的不可解释性等等。

僵尸网络技术是建立在恶意软件技术之上的产物，是恶意软件的规模化应用。攻击者可以通过僵尸网络提供的网络资源发送各种类型的网络攻击进行敲诈勒索，可以通过构建在物联网上的僵尸网络深刻影响与人们生活紧密结合的物联网设备造成公众的财产损失以牟取利益。有众多的恶意攻击者、黑客，在利益的驱使下，利用僵尸程序，与防御人员斗智斗勇，并且随着网络安全技术的进步不断地改进僵尸网络，造成了现在僵尸网络检测领域防御人员与攻击者不断竞争、相互促进进步的平衡态势。

从最近十年的发展状况来看，僵尸网络的技术发展尚未出现明显的革新。僵尸网络使用的通信协议、组织结构、逃避方式，未出现革命性创新。通信协议使用IRC、HTTP、P2P或自定义的通信协议，组织结构使用中心化、分布式、混合式结构，逃避方式主要有隐藏中心主机的Domain-Flux与Fast-Flux技术，以及僵尸程序用来隐藏自身的一系列方式，也已经有一段时间没有取得技术进展。僵尸网络技术更多是的随着互联网在人们生活中的应用，向多种不同类型的网络中进行渗透，对人们的生活产生方方面面的影响。

移动设备是恶意攻击者部署僵尸网络的新宠。随着移动互联网的发展，手机逐渐成为人们生活中不可或缺的工具，在人们的日常生活中扮演着越来越重要的角色。一位在中国的网民可能没有一台电脑，但是一定有一台手机。有相当一部分的手机用户会在手机上完成支付、转账、交通出行、办理政府业务、统计疫情信息等任务，如果手机上安装了类似僵尸程序的恶意软件，其可以组成的僵尸网络规模将是十分巨大的，相对于电脑主机端的僵尸程序，在移动设备端的僵尸程序造成的危害将更加深刻。比如最近几年出现的WireX僵尸网络，该僵尸网络由成千上万的安卓智能手机，在两个月的时间里，僵尸网络节点达到了7万规模，该僵尸网络用来发动大规模的DDOS攻击。Android系统的开放性，给了恶意软件很大的生存空间。在Google Play Store上包含WireX恶意代码的应用众多，涵盖了媒体、视频播放器、手机铃声、存储管理工具等多个类别，这也是其传播迅速的原因。该僵尸网络最终被Akamai, CloudFlare, Flashpoint, Google, Oracle Dyn, RiskIQ, Team Cymru安全公司的研究人员共同清除。针对移动设备的大规模僵尸网络危害巨大，但也不必过于担心。在移动设备上感染僵尸程序要比PC端感染僵尸程序难得多。一方面在客户端主机上没有运行通用服务，攻击者难以通过漏洞利用的方式使其感染僵尸程序，另一方面使用安卓系统的智能手机普通应用没有ROOT权限，恶意软件无法注册为系统服务，导致这些软件隐藏自身的能力有限，容易被安装在智能手机上的防护工具检测出来。移动互联网安全性也在发展之中得到了越来越多的重视，各个手机厂商也纷纷与安全厂商合作，开发了针对自己手机系统的防护工具，保障移动设备的应用安全。最近几年也有研究者提出了多种针对安装了Android系统的移动智能设备的僵尸网络检测方案。不管是移动设备组成的僵尸网络，还是PC设备构建的僵尸网络，其网络层面的行为是相似的，针对网络层面的众多检测方案对两种设备组成的僵尸网络具有相同的检测效果。此外，针对移动设备的僵尸网络检测方案的主要关注点在Android系统恶意软件识别上。研究者反编译恶意软件与正常软件的可执行文件后，提取恶意软件与正常软件的特征，随后使用各种机器学习模型与神经网络构建检测模型，实现移动设备恶意软件的检测。比如Alqatawna[38]等人2021年提出的基于静态程序分析与机器学习模型的Android系统僵尸网络检测方案。检测恶意软件的另一种方式是基于恶意软件行为，有些研究者基于软件的系统调用序列来判断该软件是否具有恶意行为，首先研究者在沙盒中运行恶意软件，监控并记录该软件的系统调用序列，使用该序列作为特征数据，结合正常的软件系统调用序列，训练机器学习与神经网络模型，实现恶意软件检测。Lv S H[39]等人在2018年提出的入侵检测方案，就是基于应用程序的系统调用序列，来实现恶意软件检测的。Android系统僵尸网络防御是一个长远的任务，我们应该积极的在智能设备上植入安全防护服务、提高用户的移动互联网安全意识，从多方面防御僵尸网络在移动智能设备的扩张。

与智能手机设备一起，物联网设备与工业设备也是僵尸网络的重要目标。物联网技术在人们日常生活中的渗透越来越深入，对人们生活的影响也越来越大。但是有众多的物联网设备在部署之初，由于网络安全意识薄弱，甚至会有裸奔（未部署任何安全防护措施）的情况，为恶意软件的传播提供了可乘之机。Maria僵尸网络是首个大规模的IOT僵尸网络案例，Maria在2016年末诞生，在不到一年的时间里传播达到了270万的规模，并在DDOS攻击中产生了1.5TB级别的僵尸流量，影响十分深远。物联网设备的安全性是能否有效防御该种类型僵尸网络的关键，物联网设备操作系统也应当具有完备的安全防御体系，以抑制甚至消除物联网僵尸程序的传播扩散。与物联网设备相比，工业设备具有更高级别的安全性，如果遭受恶意软件入侵，将造成更加严重的后果，会造成经济损失甚至人员伤亡。一般来说，因为其重要性，高等级的工业设备被入侵的可能性很小，但也不是没有可能，在工业设备这种完全不能出错的系统中，系统自身的安全性对防御僵尸程序入侵更加重要。

虽然随着人工智能技术在网络安全领域的应用，在众多研究者的投入下，僵尸网络检测技术也取得了很大的发展，但是僵尸网络检测问题仍然不能在单个维度的数据分析下解决，大规模僵尸网络的防御与反制离不开各国安全公司的信息整合、互帮互助。在防御反制WireX僵尸网络的过程中，也是有众多知名安全团队的参与，才成功捣毁了该僵尸网络。僵尸网络检测需要多维数据的参与，目前使用多维度数据检测僵尸网络的系统并不多，BotHunter作为使用多维信息检测僵尸网络的始祖，其检测能力在今天仍然是可圈可点的，这体现了多角度信息的组合对僵尸网络检测的积极作用。未来僵尸网络检测可能会发展成一个全方位的检测系统，针对移动设备、PC、物联网设备的恶意软件检测工具，针对局域网网关、网络边界网关、核心网网关的网络流量检测技术相结合，实现僵尸网络的检测与反制。针对线上的服务器集群，可以部署更加严格的恶意软件检测工具、更加高效的入侵检测技术，搭配僵尸网络流量检测能力，利用多维数据实现僵尸网络的检测与反制。随着网络安全技术的普及，物联网设备与移动设备安全性也会逐步得到提高，在这两种设备上的僵尸网络的生存会愈发困难，通过多维数据的整合，精准检测僵尸网络的目标并不是不可能实现。

参考文献

1. Carl L, Walsh R, Lapsley D, et al. Using machine learning technliques to identify botnet traffic[C]//Local Computer Networks, Proceedings 2006 31st IEEE Conference on. IEEE. 2006.
2. Garcia S, Grill M, Stiborek J, et al. An empirical comparison of botnet detection methods[J]. computers & security, 2014, 45: 100-123.
3. Beigi E B, Jazi H H, Stakhanova N, et al. Towards effective feature selection in machine learning-based botnet detection approaches[C]//2014 IEEE Conference on Communications and Network Security. IEEE, 2014: 247-255.
4. 江健, 诸葛建伟, 段海新, 等. 僵尸网络机理与防御技术[D]. 中国科学院软件研究所| 中国计算机学会, 2012.
5. Mesnards N G, Hunter D S, Hjouji Z, et al. Detecting Bots and Assessing Their Impact in Social Networks[J]. arXiv preprint arXiv:1810.12398, 2018.
6. 诸葛建伟, 韩心慧, 周勇林, 等. 僵尸网络研究[D]. , 2008.
7. 方滨兴, 崔翔, 王威. 僵尸网络综述[J]. 计算机研究与发展, 2011, 48(8): 1315.
8. Yen T F, Reiter M K. Traffic aggregation for malware detection[C]//International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, Berlin, Heidelberg, 2008: 207-227.
9. Binkley J R, Singh S. An algorithm for anomaly-based botnet detection[J]. SRUTI, 2006, 6: 7-7.
10. Alauthman M, Aslam N, Al-Kasassbeh M, et al. An efficient reinforcement learning-based Botnet detection approach[J]. Journal of Network and Computer Applications, 2020, 150: 102479.
11. Wang W, Shang Y, He Y, et al. BotMark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors[J]. Information Sciences, 2020, 511: 284-296.
12. Antonakakis M, April T, Bailey M, et al. Understanding the mirai botnet[C]//26th USENIX security symposium (USENIX Security 17). 2017: 1093-1110.
13. Bursztein E. Inside mirai the infamous iot botnet: A retrospective analysis[J]. Elie Bursztein’s site. https://www. elie. net/blog/security/inside-mirai-theinfamous-iot-botnet-a-retrospective-analysis (accessed May 26, 2020), 2017.
14. <https://tisi.org/16643>
15. <https://en.wikipedia.org/wiki/Eggdrop>
16. Arce I, Levy E. An analysis of the slapper worm[J]. IEEE Security & Privacy, 2003, 1(1): 82-87.
17. 杜勤, 吕光丽, 苏忠, 等. P2P 僵尸网络研究与进展[D]. , 2011.
18. Livadas C, Walsh R, Lapsley D, et al. Usilng machine learning technliques to identify botnet traffic[C]//Proceedings. 2006 31st IEEE conference on local computer networks. IEEE, 2006: 967-974.
19. Yen T F, Reiter M K. Traffic aggregation for malware detection[C]//International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, Berlin, Heidelberg, 2008: 207-227.
20. Choi H, Lee H, Lee H, et al. Botnet detection by monitoring group activities in DNS traffic[C]//7th IEEE International Conference on Computer and Information Technology (CIT 2007). IEEE, 2007: 715-720.
21. Gu G, Zhang J, Lee W. BotSniffer: Detecting botnet command and control channels in network traffic[J]. 2008.
22. Gu G, Perdisci R, Zhang J, et al. Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection[J]. 2008.
23. Zhuang L, Dunagan J, Simon D R, et al. Characterizing Botnets from Email Spam Records[J]. LEET, 2008, 8: 1-9.
24. Goebel J, Holz T. Rishi: Identify Bot Contaminated Hosts by IRC Nickname Evaluation[J]. HotBots, 2007, 7(8-8): 192.
25. Strayer W T, Walsh R, Livadas C, et al. Detecting botnets with tight command and control[C]//Proceedings. 2006 31st IEEE Conference on Local Computer Networks. IEEE, 2006: 195-202.
26. 邹福泰, 谭越, 王林, 等. 基于生成对抗网络的僵尸网络检测[J]. 通信学报, 2021, 42(7): 95-106.
27. 陈连栋, 张蕾, 曲武, 等. 种分布式的僵尸网络实时检测算法[J]. 计算机科学, 2016, 43(3).
28. 李可. 基于行为分析的僵尸网络对抗技术研究[J]. 北京邮电大学, 2017.
29. Gu G, Porras P A, Yegneswaran V, et al. Bothunter: Detecting malware infection through ids-driven dialog correlation[C]//USENIX Security Symposium. 2007, 7: 1-16.
30. Almutairi S, Mahfoudh S, Almutairi S, et al. Hybrid botnet detection based on host and network analysis[J]. Journal of Computer Networks and Communications, 2020, 2020.
31. Kirubavathi G, Anitha R. Botnet detection via mining of traffic flow characteristics[J]. Computers & Electrical Engineering, 2016, 50: 91-101.
32. Zhao D, Traore I, Sayed B, et al. Botnet detection based on traffic behavior analysis and flow intervals[J]. computers & security, 2013, 39: 2-16.
33. Alauthaman M, Aslam N, Zhang L, et al. A P2P Botnet detection scheme based on decision tree and adaptive multilayer neural networks[J]. Neural Computing and Applications, 2018, 29(11): 991-1004.
34. Draper-Gil G, Lashkari A H, Mamun M S I, et al. Characterization of encrypted and vpn traffic using time-related[C]//Proceedings of the 2nd international conference on information systems security and privacy (ICISSP). sn, 2016: 407-414.
35. Alauthman M, Aslam N, Al-Kasassbeh M, et al. An efficient reinforcement learning-based Botnet detection approach[J]. Journal of Network and Computer Applications, 2020, 150: 102479.
36. Sriram S, Vinayakumar R, Alazab M, et al. Network flow based IoT botnet attack detection using deep learning[C]//IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2020: 189-194.
37. Yang W, Cheng Z, Cui B. Recombining TCP sessions based on finite state machine to detect cyber attackers[C]//Proceedings of the 3rd International Conference on Cryptography, Security and Privacy. 2019: 138-142.
38. Alqatawna J, Ala’M A Z, Hassonah M A, et al. Android botnet detection using machine learning models based on a comprehensive static analysis approach[J]. Journal of Information Security and Applications, 2021, 58: 102735.
39. Lv S H, Wang J, Yang Y Q, et al. Intrusion prediction with system-call sequence-to-sequence model[J]. IEEE Access, 2018, 6: 71413-71421.

致谢

本论文是在导师的悉心指导下完成的，从论文的选题到论文的撰写，无不渗透着导师的心血，……值此论文完稿之际，谨对导师的辛勤培育以及谆谆教诲表示最衷心的感谢!

作者简介

##### 基本情况

杨智乐，男，河南洛阳人，1996年8月出生，西安电子科技大学信息安全学院网络空间安全专业2019级硕士研究生。

##### 教育背景

2015.08～2019.07西安电子科技大学，本科，专业：通信工程

2019.08～ 西安电子科技大学，硕士研究生，专业：网络空间安全

##### 攻读硕士学位期间的研究成果

###### 发表学术论文

无

###### 申请（授权）专利

无

###### 参与科研项目及获奖

1. 项目, 项目名称, 起止时间, 完成情况, 作者贡献。