2019 年全国大学生信息安全竞赛

作品报告

作品名称： 基于可信度的网络威胁情报繁殖系统

电子邮箱： 1347542031@qq.com

提交日期： 2019 年 5 月 24 日

填写说明

1. 所有参赛项目必须为一个基本完整的设计。作品报告书旨在能够清晰准确地阐述（或图示）该参赛队的参赛项目（或方案）。

2. 作品报告采用A4纸撰写。除标题外，所有内容必需为宋体、小四号字、1.5倍行距。

3. 作品报告中各项目说明文字部分仅供参考，作品报告书撰写完毕后，请删除所有说明文字。(本页不删除)

4. 作品报告模板里已经列的内容仅供参考,作者可以在此基础上增加内容或对文档结构进行微调。

5. 为保证网评的公平、公正，作品报告中应避免出现作者所在学校、院系和指导教师等泄露身份的信息。

目录

摘要

“信息全球化”的浪潮为网络空间安全的地位的提升带来了极大的契机，网络空间安全已成为国家安全体系的重要组成部分，可谓是牵一发而动全身。然而，层出不穷的网络空间安全威胁也成为了人们不得不面对的新挑战。现有的威胁情报分析系统在威胁情报获取、处理和利用以及分析成果展示方面有诸多问题亟待解决，比如威胁情报不能实时获取、原有的机器学习模型较为单一、模型应用效果不好等，导致目前还没有出现高效有效的威胁情报分析系统。

本作品提出了基于可信度的网络威胁情报繁殖系统，这个系统可以动态地获取威胁情报，并且对未知的情报进行分析和判断，利用基于可信度的多模型融合方法动态地判定它们的威胁性，将判定为威胁的情报加入训练集中，扩展现有的威胁情报库，用以应对现有的和将来可能面临的网络威胁。

本系统特色如下:

一、基于多模型的海量威胁情报分析处理手段，建立了情报预测处理模块，缓解了情报预测的不准确和不全面的问题。

现有的分析系统基本上采用的是单一的机器学习模型，本系统采用多种机器学习模型。不同的机器学习模型对于不同的威胁情报的拟合程度不同，单一的机器学习模型可能会和某些威胁情报的拟合程度不高，可能会存在一些漏网之鱼或者预测出错的情况，但混合模型就能在一定程度上缓解这个问题。本系统利用多种异构的机器学习算法，对系统获取到的海量威胁情报数据进行建模并对模型打分。多模型的利用提高了情报预测的准确性，也有利于算法间的互相优化和机器学习。

二、基于可信度统计算法的威胁情报分析系统架构，突破基于阈值的分析方式， 提高情报利用率。

现有的分析系统通常采用传统的、基于阈值的机器学习模型，而这种机器学习模型在安全领域的应用效果不太好。基于阈值的机器学习模型根据一个固定的临界值对威胁情报进行判断分类，在面对不断变化的威胁情报时，一个固定的临界值无法满足判断情报的需求，很可能出现新的、在阈值之外无法判断的威胁情报；而本系统采用的基于可信度的机器学习模型在大数据的条件下，根据实时更新的威胁情报，能够提供一个动态的置信区间。静态和动态的数据分析在面对与时俱进的威胁情报时，效果截然不同，很显然动态的置信区间更具有优越性。

三、基于流量的实时监测和大数据分析，建立了大数据威胁情报实时展示平台， 提升了威胁情报的利用效果。

本系统对用户访问的流量进行实时的检测和分析，并且对计算出的可信度进行了平台化的可视化展示，更加人性化，普遍化，实用性更强。 系统测试结果表明，本系统充分挖掘海量威胁情报的价值，建立多个异构机器学习模型，并在预测结果上，进行二次基于可信度的统计学习（利用Inductive Venn-Abers Predictive Distribution），繁殖出新的威胁情报，为威胁情报的共享和协同防御提供了新的思路。

经实验证明，该方法能够更有效地进行对威胁情报的识别和防御。

关键词：威胁情报；统计学习；机器学习；实时可视化；IVAPD

第一章 作品概述

1.1 背景分析

信息时代，网络空间已成为陆、海、空、天之外人类活动的“第五空间”。政治、经济、文化、社会、军事等国家重要领域的基础设施与网络空间联系日益紧密，网络安全对国家安全牵一发而动全身，已成为国家安全体系的重要组成部分。“信息全球化”的浪潮为网络空间安全的地位的提升带来了极大的契机，然而层出不穷的网络空间安全威胁也成为了人们不得不面对的新挑战。急速发展的针对性网络攻击直接催生了威胁情报服务。威胁情报分析作为网络安全领域的热点问题，能够让威胁更加清晰可见，更快速响应针对性攻击，加强策略规划和投资，同时有效缓解目前在对抗网络攻击时的攻防不对等问题，也为网络态势实时感知提供了技术支持。 近年来，我国网络空间安全威胁数量激增，而我国对威胁情报的分析仍存在处理灵活性不足、利用率不高、应用性不强等特点，相关研究亟待深入。

1. 我国网民被动受害，缺乏信息预测

互联网时代下网民数量激增，人们与互联网的联系越来越紧密。

2019 年 2 月 28 日，中国互联网络信息中心（CNNIC）发布了第 43 次《中国互联网络发展状况统计报告》。如图 1-1 显示，截至 2018 年 12 月，中国网民的规模达到了 8.29 亿，全年新增网民的数量是 5653 万，互联网的普及率是 59.6%，较前年底提升了 3.8 个百分点；中国手机网民的规模达到了 8.17 亿，全年新增手机网民的数量是 6433 万。截止去年 12 月，我国即时通信用户规模达 7.92 亿，网络新闻用户规模达 6.75 亿，网络购物用户规模达 6.10 亿，网上外卖用户规模达 4.06 亿，网络支付用户规模达 6.00 亿，网络视频用户规模达 6.12 亿，短视频用户规模达 6.48 亿。

图 1-1 中国国民规模 [1]

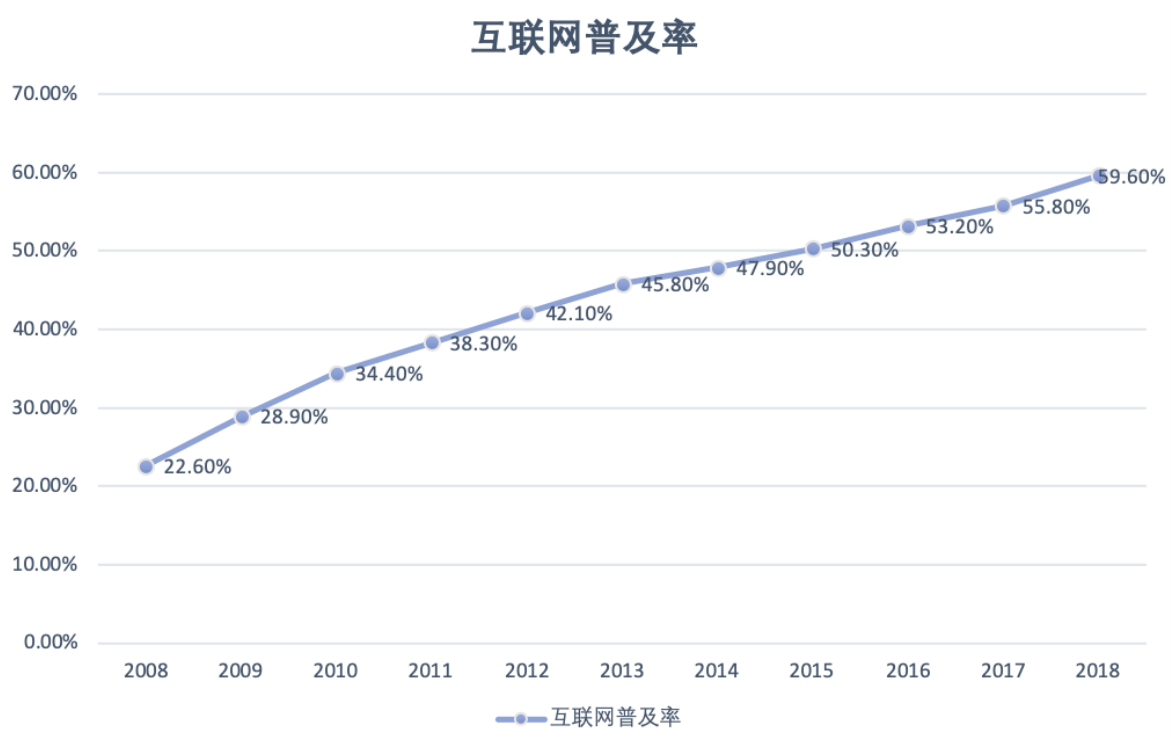

图 1-2 中国互联网普及率 [2]

随着信息技术的发展，互联网已经成为我们生活中不可缺少的重要组成部分。

[1]图源 CNNIC 发布的第 43 次《中国互联网络发展状况统计报告》
[2]图源 CNNIC 发布的第 43 次《中国互联网络发展状况统计报告》

但是在如此庞大的网络用户量背后，却潜藏着极大的安全隐患。 如今，新一代的攻击者常常向企业和组织发起针对性的网络攻击，造成了巨大的网络威胁，近年来我们经常能听到各种由于网络安全问题导致的网络用户利益受损的事件：

2017 年 5 月，WannaCry 勒索病毒全球大爆发，至少 150 个国家、30 万名用户中招，造成损失达 80 亿美元，已经影响到金融，能源，医疗等众多行业，造成严重的危机管理问题。中国部分 Windows 操作系统用户遭受感染，校园网用户首当其冲，受害严重，大量实验室数据和毕业设计被锁定加密。部分大型企业的应用系统和数据库文件被加密后，无法正常工作，影响巨大。

2019 年 1 月， MEGA 上泄漏的的一个容量超过 87GB 的公开数据集中包含了 7.73 亿电子邮件地址和 2122 万个唯一密码。这个庞大的数据量，使其成为有史以来载入 HIBP 网站的最大的漏洞。

而网络安全案例不止如此，最近几个备受关注的案例包括：

网络犯罪者向零售商、银行和其他组织发起针对性攻击，以获得经济利益；“激进黑客”（hacktivist）和国家背景的黑客攻击媒体、金融组织、政府机构，以实现政治目的。其他的案例包括：私营或国营企业盗取国防企业及制造商的工程和业务流程信息；懂金融的黑客攻击医疗和制药公司，来获取影响股票价格的内部信息。

这些黑客不断改变现有的攻击方式，开发新的方法，单独依赖防火墙、入侵防御系统和反病毒软件已经无法阻止这些黑客的攻击。这类攻击无法通过恶意程序签名或者过去的攻击技术报告进行检测。而且，实际上，大多数企业面临的现状是收到的原始威胁数据过多：有太多警报，太多漏洞预警和补丁，太多关于各类恶意软件、钓鱼攻击和 DDoS 攻击的报告。

而不仅是政府和企业层面普通网络用户也面临着巨大的网络安全威胁：网络用户在上网的过程中会面临病毒攻击、误访问不安全的网站，还会遇到很多被篡改的网站，或者被植入后门的网站，甚至一些政府网站也无法幸免。

急速增长的针对性网络攻击直接催生了威胁情报分析。捕获威胁情报对其进行分析和预测，能够有效缓解目前这种攻防不对等的情况，进一步保障网络空间的安全。并且，威胁情报的检测需要更加的精准和有效，才可以防止这些网站对用户造成欺骗，从而造成用户的财产损失或者信息泄露等情况；而威胁情报的繁殖则需要更加的准确和及时，才可以更多、更早地预测到新的威胁情报，提前避免有可能造成的损失。

2. 威胁情报时效性显著，开源情报容易冲突

目前无论是工业界还是学术界对威胁情报都还没有一个统一的定义，许多机构或论文都对威胁情报的概念进行过阐述，目前接受范围较广的是 Gartner 在 2014 年发表的《安全威胁情报服务市场指南》（Market Guide for Security Threat Intelligence Service）中提出的定义，即："威胁情报是关于 IT 或信息资产所面临的现有或潜在威胁的循证知识，包括情境、机制、指标、推论与可行建议，这些知识可为威胁响应提供决策依据。"

我们在这里主要采用以上这种定义。

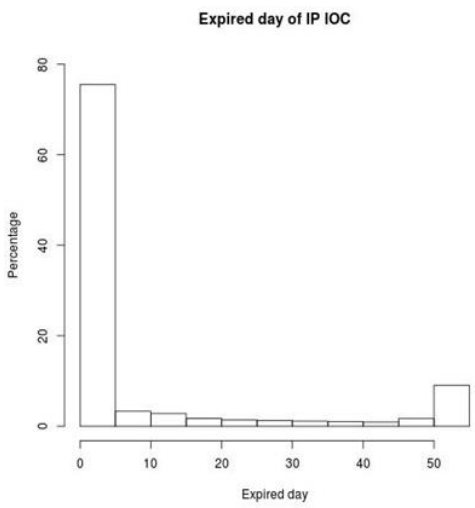

图 1-3 IP 恶意情报的持续时间

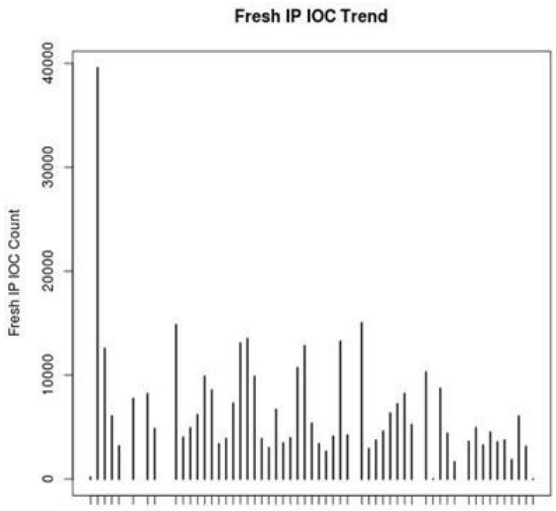

图 1-4 每天新增的新 IP 恶意情报数量

时效性强是威胁情报的重要特点。如图 1-3 和图 1-4，这是来自绿盟科技创新中心 2018 年统计的数据，分别是 IP 恶意情报的存活时间和每天新的 IP 恶意情报数量。我们可以从图中看出 75%的恶意 IP 情报持续时间在 5 天内，平均每天 6668 个新增 IP IOC(部分外部采集源)。而时效性也是开源情报的一个重要问题。

许多开源情报往往并没有标注持续时间，仅标记生成时间。很多都是每天一个列表，应用者只知道开源情报平台什么时候发现该恶意对象，并不知道该对象是否持续作恶。情报的域名拥有者、IP 使用者和其上的业务，随着时间的推移，可能产生变化，黑 IP 会变成白 IP。过时或失真的情报会在实际使用中给应急处置带来大量的垃圾警告，给安全管理人员造成困扰。因而，光靠采集外部情报进行威胁情报平台建设往往

有着数据有效性上的怀疑。

由于威胁情报本身存活时间很短，再加上每日会不断的出现大量新情报，因此，威胁情报的繁殖十分具有现实意义。进行威胁情报繁殖，可以及时的扩容威胁情报数据库，让情报库的更新紧跟威胁情报的更新迭代，让预测系统对新出现的威胁情报有很好的预测效果。

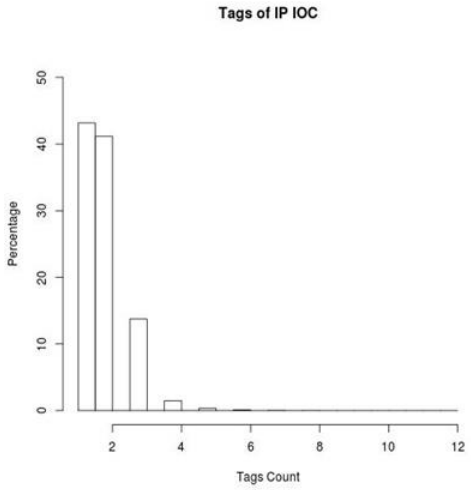

图 1-5 多源多标签标记

还是根据绿盟统计的开源情报数据（图 1-5），可以看到 57%的恶意 IP 情报被标记多个类型或被多个情报源标记。三人成虎的方法往往是威胁情报业界的基本做法，即多个来源说一个 IP 是恶意的，它就更恶意（恶意置信度或威胁指数上升），但其实这里有个现实的逻辑陷阱：由于无从考证开源情报源的基础数据来源，所以无法得知各个情报源之间是否有相互“抄袭”的状况。如果单纯三人成虎，则很有可能产生循环论证的现象，进而导致情报源的可信度下降。

同时，开源情报不仅有黑情报，还有白 IP 情报，业界往往把不同维度上访问量高的 IP 和域名作为可信的白名单，例如思科的 Umbrella Popularity List 和 Alexa 的 Top1m List。如果将域名对应的 Alexa 排名赋予其指向的 IP，然后和黑名单 IP 关联比较，可以看出，即使强如 Alexa 排名前一百的 IP，冲突数量也有数百。通常情况下，企业外发流量往往 70%是访问 Alexa 排名前一百万的。这就意味着如果拿开源情报在企业实时流量中匹配会产生大量的误报。图 1-6 纵轴是冲突数量，横轴是 Alexa 的排名区间。

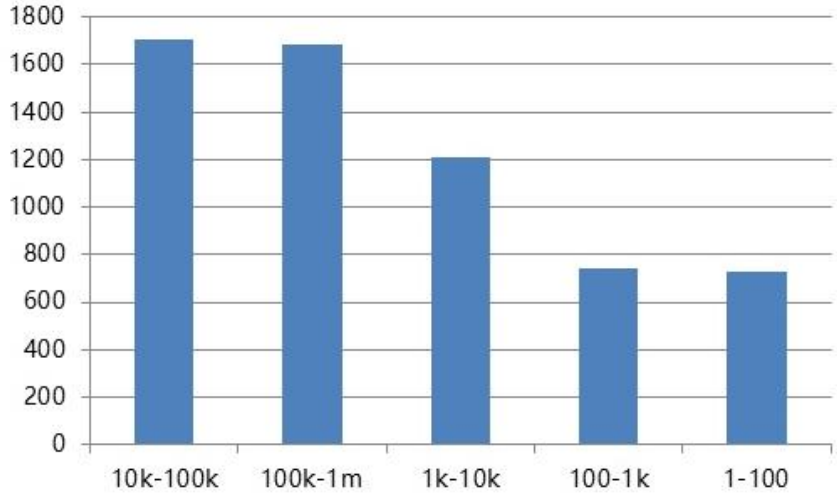

图 1-6 多源多标签标记

针对这一现象，威胁情报的繁殖则可以确定数据的不重复性，对繁殖出的新威胁情报做到和库中情报不同并且保证预测可信度极高，从而保证各采用团队使用过程中的数据准确性和合理性。

3.全球对威胁情报的应用需求激增

SANS 在 2019 年 2 月发布了《SANS 网络威胁情报现状调研报告》（The Evolution of Cyber Threat Intelligence （CTI）: 2019 SANS CTI Survey） SANS 选取了全球 220 个企业，涉及 IT、政府、银行和金融、制造、教育、医疗、咨询和通信等行业。其中规模超过 5000 人的企业占比 48%。

根据最新发布的 2019 年报告，总体上，SANS 认为 CTI（网络威胁情报）的应用越发成熟，其发挥的价值也越来越大，CTI 的应用正逐步深化。

1）报告显示，72%的受访组织生产或消费了 CTI，比 2017 年的 60%有显著提升。（如图 1-7）

2）更多的组织开始关注威胁情报

3）81%的受访者认为 CTI 对于安全阻断/检测/响应是有价值的（如图 1-8）

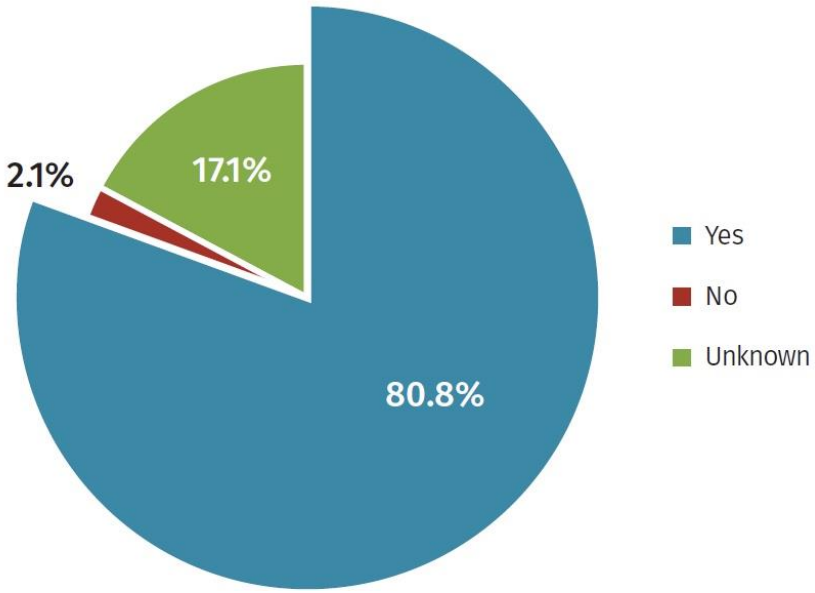

图 1-7 是否生产或消费了 CTI[3]

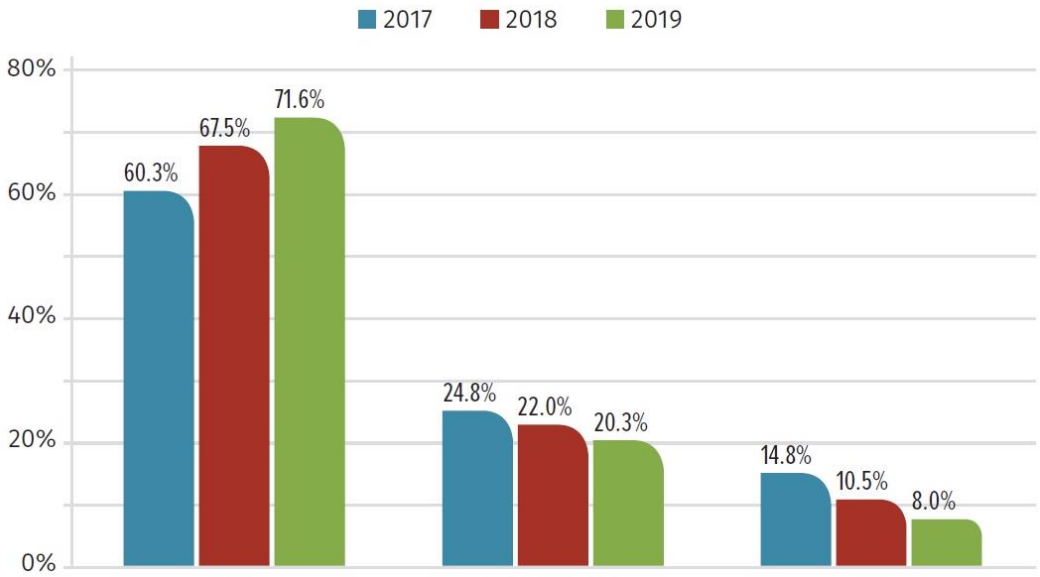

图 1-8 CTI 对于安全阻断/检测/响应是否有价值的[4]

根据这一报告，我们可以得出随着威胁情报应用需求的激增，对于威胁情报的收集和分析以及繁殖是十分有必要的。

[3] SANS 在 2019 年 2 月发布的《SANS 网络威胁情报现状调研报告》(The Evolution of Cyber Threat Intelligence （CTI）: 2019 SANS CTI Survey)

[4] SANS 在 2019 年 2 月发布的《SANS 网络威胁情报现状调研报告》(The Evolution of Cyber Threat Intelligence （CTI）: 2019 SANS CTI Survey)

4. 威胁情报深度利用需求显现，相关研究还需完善

2015 年 2 月 10 日美国政府宣布成立网络威胁与情报整合中心，该机构工作是协调整合国土安全部、联邦调查局等多部门的情报力量，提供及时，有见地，客观和相关的情报，为国家安全问题和事件的决策提供信息，提高美国防范和应对网络攻击的能力。

相比之下，我国对威胁情报的深度利用工作较为匮乏，威胁情报的分析还达不到提供精准决策的程度，这对于我们维护自身网络空间安全时极为不利的。

同时对于国内的普通网络用户而言，目前的恶意网站的预测拦截不能提供可信度，在访问网站时无法进行准确高效的决策，因此一些用户仍然可能在无法得知可信度的情况下访问恶意网站，这也造成巨大的网络风险，有可能带来一定程度的经济损失。

传统的基于阈值的威胁情报分析只能做到 0 和 1 的区分，即通过与国内、外安全公司、安全组织公开发布的黑名单对比只能确定该情报是绝对的无害的或者是绝对的有害的，但是却并不能对其进行进一步的分析，不能将该情报的价值完全地发掘与利用。我们认为，威胁情报分析的目的不应只是反映当前的网络安全状况，更应根据收集到的纵向历史数据和当前数据，对未来网络安全威胁态势的发展做出较为准确且有意义的预判；同时, 根据横向的同类、多源、异构、分布式和内外来源兼有的威胁情报数据,对未来威胁态势的扩散范围、影响等做出分析。

5. 威胁情报大数据，统计学习新趋势

随着网络的发展、威胁情报数量的激增，传统的情报分析方式已经无法适应其变化，无论是战略型、战术型还是运营型威胁情报分析方式，都必须实现对海量的原始数据的充分利用，提取真正有用的网络安全威胁信息，建立高效智能的信息对比算法，实施多源数据复杂关联、快速检索与情报跟踪等都需要引入新理念。

2012 年 3 月，美国政府发布《大数据研究和发展倡议》，启动了大数据的战略布局，数据驱动的环境和发展趋势逐渐形成。大数据具备的 4V 特性：Variety，多类型数据格式支持；Volume，大容量存储;Velocity，快速处理；Value，数据价值高带给网络安全领域启发性变化。

大数据分析技术的应用将使针对攻击行为的主动监测性增强，原本孤立的威胁信息得到有效整合，从而繁殖出更多的信息，在对关联数据的深度分析中使得数据价值充分得到挖掘，利用率显著提升。并且基于统计学习的威胁情报繁殖，使得威胁情报

不断更新，为网络态势实时感知及有效预测提供技术支持。

1.2 相关工作

1.威胁情报在国内外的研究综述

美国在 20 世纪末就开始关注网络威胁情报，发展到目前美国的威胁情报市场已比较成熟。

IBM 公司推出的 X-Force 平台，基于超过 20000 台的托管设备，每天管理 133 个国家/地区超过 150 亿个事件，拥有超过 2.7 亿个端点报告恶意软件;基于以上分析范围，IBM X- Force 在深度方面则可以做到分析超过 250 亿个网页和图像、每天超过 1200 万垃圾邮件和网络钓鱼攻击、以及超过 86 万个恶意 IP 地址。而通过 X-Force 威胁平台与安全产品的结合，可以实现实时的 IP、漏洞、URL 分析，并可根据威胁情报 设定企业安全阈值，并获取最新安全信息的持续更新。

Anomali 公司推出的 ThreatStream 威胁情报平台，汇聚了数以万计的威胁情报信息来识别新的攻击，并发现已知的漏洞，使安全团队能够快速发现并阻止相关威胁。除此之外还有 Virustotal、Threatcrowd 等。

国内关于威胁情报的研究态势良好，并已经建立了许多的威胁情报开放平台，如绿盟威胁分析中心、360 威胁情报中心和微步在线等。目前 360 公司的 360 威胁情报中心可通过集成多方的威胁情报和基础网络数据，使安全人员可以对报警有比较明确的判断，具体方式如：相关域名、IP 历史上是否被发现恶意攻击行为；域名和 IP 相关的已知的恶意软件；访问来源是否可疑（如:IDC 服务器作为终端来访问 Web 应用，通过 Tor、 VPN、代理的访问）等。 国内做的最好的应该是微步在线，微步不仅仅只有一个搜索功能，还建立了相应的社区进行情报共享，并实现了威胁情报的产品化。

这些威胁情报平台一般都以一个“搜索引擎”作为入口，输入 IP、域名、文件 HASH 等信息输出查询的结果，包括杀毒引擎检测率、Whois、PDNS、关系图谱等。有的也会提供 API 接口，或者提供情报推送服务，并作为一种盈利方式。

2. 机器学习在威胁情报中的研究

大数据时代下的威胁情报数据庞杂，传统人工分析速度慢、处理数据量小，加之成本高，已经难以满足实际需求。而目前的恶意攻击又具有高度的重复性，如果利用机器学习来处理威胁情报，能够加快处理海量数据的速度，同时机器学习又具有成本

低、自动化等特点，弥补人工分析的缺陷。

目前，大多数多数威胁情报分析平台都在使用机器学习为其提供强有力的技术支持。比如，面对海量且不断复杂化的安全威胁，Fortinet 威胁防御系统采用了 AutoCPRL Signature(自动内容分析)技术，Auto-CPRL 可以通过机器学习生成安全特征，通过单一类别特征识别一个家族的恶意软件，分析速度 200 倍速度于人类分析，可以有效检测到多种形态的恶意软件家族的成员。

而国内首家以威胁情报服务为中心的公司微步在线研发的产品 TDP ，是采用基于机器学习的检测引擎在各种流量攻击下自动训练和改进系统，能够更好地发现 DGA、异常连接和数据泄露。

3. 传统预测模式的缺陷

目前，大多数威胁情报分析平台的反馈结果是绝对的，即该信息完全是恶意的或者完全是正规的，并没有充分挖掘恶意数据背后的信息。包括现在很多浏览器已经能够实现对恶意网站的拦截，但是并未给出这个网站威胁性的可信度，而只是告知用户这个网站可能是恶意网站。因此，如果能够改进当前利用威胁情报的预测结果，增加可信度评估以摆脱固定阈值的限制， 实现威胁情报的充分利用，将会更有助于我们改善网络用户体验，进一步提升网络安全防护水平。

1.3 作品简介

威胁情报，也被称作安全情报、安全威胁情报，通过大数据、分布式系统或其他特定收集方式获取，包括漏洞、威胁、特征、行为等一系列证据的知识集合及可操作性建议，可还原已发生的和预测未来可能发生的网络攻击，为用户决策提供参考依据，帮助用户避免或减小网络攻击带来的损失。威胁情报技术的出现和发展被看作是解决今后网络安全问题的关键。

本系统是一种基于可信度的网络威胁情报繁殖系统。本系统利用编写的 Python 爬虫，从全球 60 多个安全公司、安全组织的相关网站上获取其网站公开的黑名单数据，从 Alexa 上获取到前 100 万个域名作为白名单，最终利用正则表达式进行处理和数据的合并，我们得到关于网站的白名单与黑名单，并以此建立我们的威胁情报库。用户在进行网站的访问时， 利用 Zeek/bro 对该网站的 IP、域名等进行分析，从而得到了一个新的数据，在原有的黑名单基础上加入该数据并运用多模型机器学习进行打分。根据第一次机器学习的得分，再运用 IVAPD（Inductive Venn-Abers Predictive

Distribution）对该结果进行校准，从而获得网站的可信度，并将其提供给用户，使用户知道该网址的可信程度。同时将分析后认定为危险网址的数据加入危险情报库，形成新的威胁情报库。实现了威胁情报的繁殖和进一步利用。

具体步骤如下:

1.威胁情报的收集：

利用 Python 爬虫获取原始数据，利用正则表达式处理数据，最终得到网站的黑名单与白名单，利用这些威胁情报建立威胁情报库（每天定点更新） 。

2. 威胁情报的利用：

用户进行网站访问时，使用开源的流量分析器 Zeek/bro 对该网站的行为进行分析，生成该网站相关信息的日志。之后对该日志进行处理，将 log 文件转化整合成所需要信息的 csv 文件，并将处理后数据作为机器学习的输入，利用多机器学习模型分别对这个数据进行预测，最后根据各个模型的可信度对预测的结果进行综合。

3. 威胁情报的繁殖：

利用 IVAPD（Inductive Venn-Abers Predictive Distribution），对多机器学习算法威胁情报模型进行校准，并将校准后的威胁情报并入威胁情报库，实现威胁情报的繁殖。结果进行可信度计算，将可信度高的结果提供该用户。并且根据可信度来判断是否为威胁情报，并且将可信度较高的预测结果加入情报库，实现威胁情报的繁殖。

4. 威胁情报的大数据可视化：

利用 superset 将获取的用户访问的网站的相关信息进行可视化展示，将网站的 port, sources, destinations, services, protocols, states 信息以图形的方式展示出来，使用户具体全面地了解其访问的网站的信息，同时将通过本系统得出的结果进行展示，让用户了解经过本系统进行分析后的对该网站的判断结果。

1.4 特色与创新点

1.基于流量的实时监测和大数据分析

本系统利用一种被动的开源流量分析器 Zeek/bro 对网站信息进行分析，该分析器可以对链路上所有深层次的可疑行为流量进行一个安全监控，支持在安全域之外进行大范围的流量分析，分析包括性能评估和错误定位。用户在进行网站的访问时，Zeek/bro 即对该网站的 IP、域名等信息进行分析，获得关于这个网站的日志并作为机器学习的输入，在后续过程中对该数据进行进一步分析。

所以本系统可以在用户对网站进行访问时，就对网站实时地进行分析，将获得的信息进行处理，获得包含本网站的相关信息的格式化的日志，具有可读性，便于共享，实现了对威胁情报实时的收集。

2. 多种算法对海量威胁情报进行分析处理的手段

本系统同时采用多种机器学习算法对获取的威胁情报进行分析，能够降低由于算法与威胁情报拟合程度不高带来的误差风险，较大程度上地避免错误结果的产生，大大提高了结果的准确度，选择几类“好而不同”的模型进行融合，甚至可以达到比最好的模型还要好的结果。

3. 基于可信度的威胁情报分析系统架构

本系统可以通过 IVAPD（Inductive Venn-Abers Predictive Distribution），对多种机器学习算法威胁情报模型计算其可信度并进行排名。同时，用户可以根据自己对于准确率、召回率的不同需求，选取不同的参数 k（选取的模型数）和 rate（通过率），结合可信度最高的 k 个模型得出最终的分析结果。

4. 实现威胁情报的繁殖

本系统在对获取的情报进行分析后，得到一个关于网站是否安全的分析结果，将认定为危险的网站加入威胁情报库中，更新威胁情报库，实现威胁情报的繁殖，大大提高了对威胁情报的利用率，使威胁情报的价值得到更大的发挥，更新了的情报库也更适应快速更新迭代的威胁情报，更能迅速有效地做出反应。

5. 大数据可视化展示

本系统在对获得的新情报进行分析且进行威胁情报库的更新后，使用可视化平台（superset）对数据进行可视化展示，将分析的数据进行图形化，如将生成的图形化效果展示给用户，便于用户更加直观地了解该网站的详细情况。

1.5 研究意义

“信息全球化”的浪潮为网络空间安全的地位的提升带来了极大的契机，随着移动互联网、物联网、互联网+、智慧城市、智能家居等新技术和新应用的出现，网络空间安全的广度和深度不断扩展，网络安全对抗越来越激烈和智能化，网络空间安全防御模式和技术需要不断创新。 然而层出不穷的网络空间安全威胁也成为了人们不得不面对的新挑战。

急速增长的针对性网络攻击直接催生了威胁情报服务。威胁情报研究是目前国内外网络空间安全的热点之一。基于可信度的网络威胁情报繁殖系统，能够在威胁情报研究的过程中，通过威胁情报收集，了解当前敌方对自己的威胁信息，提前做好威胁防范、检测以及响应，在网络空间安全的对抗中做到“知彼”; 同时通过威胁情报利用，将本地网络实时安全数据与海量威胁情报进行匹配，及时发现本地网络的安全威胁，在对抗中做到“知己”；并且通过威胁情报繁殖，利用机器学习算法对海量威胁情报数据进行建模，再利用 IVAPD（Inductive Venn-Abers Predictive Distribution）对模型的预测结果进行校准， 挖掘出新的本地化的威胁情报，实现威胁情报的繁殖，建立威胁情报情报共享与协同防御机制。

综上所述，能够实现网络空间安全对抗中的“知己知彼”和“协同防御”。

除此之外，该系统将威胁情报的收集、利用及繁殖实现可视化，能够使用户更加直观的得到网站信息。

基于可信度的网络威胁情报繁殖系统兼具现实安全上的意义和系统数据库繁殖上的能力，在人们生活及科学研究的各个方面具有不容忽视意义。

1.在网站访问安全方面

信息时代的网络用户日常生活中面临着许多恶意网站造成的网络安全威胁，基于可信度的网络威胁情报繁殖系统可以实时的监控用户上网的流量情况，并且针对其访问的各种网站进行威胁情报的繁殖匹配，通过给予用户可信度的方式，让其对其所即将访问的网站的安全性有一个清晰的了解，从而来规划其网络的访问并且保证其信息的安全。不同于传统的基于阈值的威胁情报分析导致的仅仅对用户做出建议禁止访问对应网站的判断；我们在威胁情报的传统模式上增加可信度的计算，将可信度反馈给用户后将判断权交给用户。根据可信度，用户可以针对自己所能接受的风险承受能力来选择是否继续访问对应的网站。从而可以让用户在自己可以接受的范围内获得更多的使用收益和体验。

2. 在学术和科研方面

利用诸多的威胁情报分析和统计学习，在判断计算可信度提供给用户的同时不断更新数据库。在由 60 多个安全公司、安全组织发布的网络恶意行为信息组成的威胁情报库的基础上，不断加入新的依据可信度判断出的威胁情报，使得数据库得到不断的更新。不断更新的数据库可以给更多类似的学术研究提供数据和可信度预测，进行

更多的大数据分析和机器学习。同时开放的系统平台可以得到更多的合作和更好的机器学习检测模型，使得其不断的优化，让其可信度计算的准确性合理性得到更好的优化。

1.6 应用前景

基于可信度的网络威胁情报繁殖系统不仅仅在对网络威胁情报的监测与分析上有诸多设计和功能，同时对于利用威胁情报分析进行网络安全防御方面也有很好的应用前景。作品突破传统的基于阈值的威胁情报分析，并且采用多机器学习模型，提高了情报利用率，缓解了情报预测的不全面和不准确的问题。本系统可以向许多潜在受众进行推广，比如:

1.互联网安全公司

本系统一方面可以为广大有网络安全需求的互联网公司提供海量威胁情报的分析和判定功能，并且为其建立大数据威胁情报实时展示平台，使得它们可以更加直观地了解威胁情报分析情况；另一方面，还可以提供通过该系统繁殖出的新的本地化的威胁情报，建立情报共享和协同防御体系。

2. 普通网络用户

本系统在涵盖了传统威胁情报分析系统的功能以外，还能提供更新更好的帮助网民抵御网络威胁的方式：本系统不但能告知网民其正在访问的网站是否存在威胁信息，同时还可以以提供威胁情报的可信度的形式，使网络用户更好的判断即将浏览的网站存在的潜在威胁是否属于其可接受范围，而不是像传统的系统那样直接劝阻其停止相关访问。

3. 政府部门

本系统可以在网络空间安全的对抗中做到“知己”和“知彼”，使得相关部门可以提前做好对网络空间威胁的防范、检测、响应，保障公民信息财产安全，维护公民合法权益，促进社会主义和谐社会的建设与社会主义精神文明建设。

4. 科研工作者

本系统成功部署和推广后，有利于威胁情报数据库的繁殖和扩大，同时也将为各类相似信息安全的科研提供数据支持。由于这一系统的灵活性和开放性，可以引入各类机器学习算法，十分适合各类信息安全领域的研究。

1.7 本章小结

在本章，我们对作品做了一个整体的描述，包括作品的研究背景、相关工作、作品简介、特色与创新点、研究意义以及应用前景等。从我国当前威胁情报分析的需求和存在的问题入手，我们就行了充分的调查、规划并且提出了相关创新点，分析了本系统的主要研究方向、方法和一些具体内容，阐述了其实现意义和前景展望，为后期作品的具体设计与实践奠定基础。

第二章 作品设计与实现

2.1 总体设计

本系统主要可以分为四大部分：信息收集（包括黑白名单的收集和本机网络行为的收集）、数据分析、机器学习、可视化展示。

信息收集模块当中的收集黑白名单部分，我们利用 python 爬虫技术对一些安全公司的网站的信息进行爬取，再将得到的原始数据集进行格式统一（正则化数据处理），本机网络行为收集利用 Zeek/bro 来对网络安全进行监视。

在数据分析模块我们对于域名进行解析，得到域名的一些特征值，为后续的机器学习模块做准备。

机器学习模块当中，我们采用的是基于可信度的多模型的机器学习，先用多个模型分别对数据集进行拟合，再使用 IVAPD 对模型进行评估打分，取可信度最高的 k 个模型组成多模型，这些模型分别对域名的安全性进行预测，最后根据用户设定的参数结合不同模型的预测结果，得到最终的结果。

最终我们对于机器学习模块的结果用 superset 进行可视化展示。

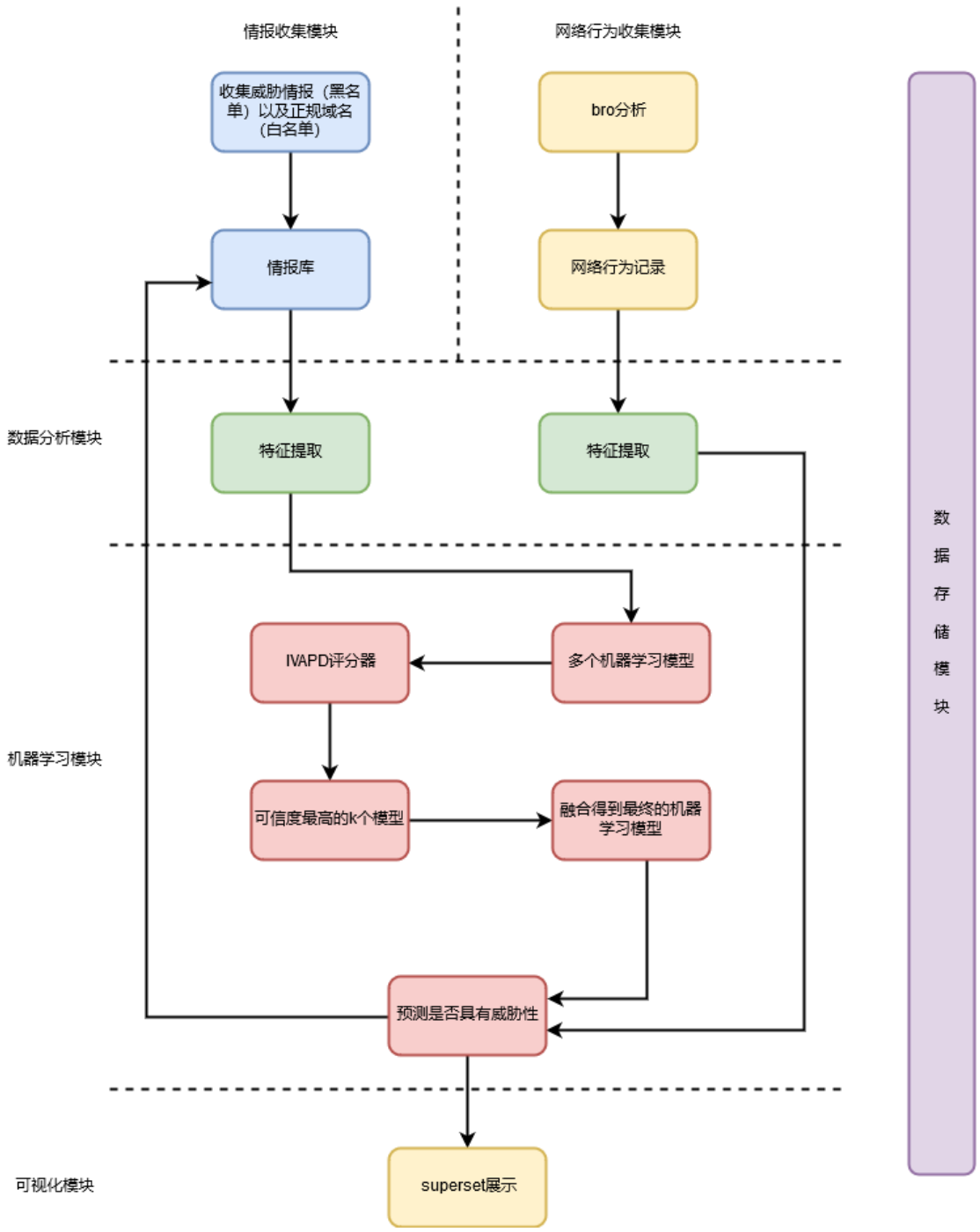

图 2-1 系统总体设计

2.2 主要依赖的python库

1. import numpy as np

numpy 是 Python 语言的一个扩展程序库，支持大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。

2. import pandas as pd

pandas 是一个开放源码、BSD 许可的库，为 Python 编程语言提供高性能、易于使用的数据结构和数据分析工具，pandas 还有很多对于 dataframe 和 csv 文件操作的函数。

3. from keras.models import Sequential

   from keras.layers import Dense, LSTM

Keras 是一个高级神经网络 API，用 Python 编写，能够在 TensorFlow，CNTK 或 Theano 之上运行。我们需要的 LSTM 模型就从这里调用。LSTM，（长短期记忆，Long short-term memory, LSTM）是一种特殊的 RNN（循环神经网络，Recurrent Neural Network），主要是为了解决长序列训练过程中的梯度消失和梯度爆炸问题。简单来说，相比较于普通的 RNN，LSTM 能够在更长的序列中有更好的表现。

4. import xgboost as xgb

XGBoost 是一个优化的分布式梯度增强库，旨在实现高效，灵活和便携。 它在 Gradient Boosting 框架下实现机器学习算法。 XGBoost 提供并行树提升（也称为 GBDT，GBM），可以快速准确地解决许多数据科学问题。 相同的代码在主要的分布式环境（Hadoop，SGE，MPI）上运行，并且可以解决数十亿个示例之外的问题。

5. from sklearn.ensemble import RandomForestClassifier

   from sklearn.ensemble import GradientBoostingClassifier

   from sklearn.ensemble import AdaBoostClassifier

   from sklearn.ensemble import BaggingClassifier

   from sklearn.linear_model import LogisticRegression

   from sklearn.naive_bayes import GaussianNB

   from sklearn.neighbors import KNeighborsClassifier

   from sklearn.linear_model import LinearRegression

scikit-learn，一个 Python 当中的机器学习库，是简单有效的数据挖掘和数据分析工具，基于 NumPy，SciPy 和 matplotlib 构建。我们需要从 sklearn 类库当中导入随机森林、朴素贝叶斯等等机器学习模型。

6. from sklearn.model_selection import KFold

我们在对模型进行验证的时候采用 k 折交叉检验，所以需要 sklearn 当中的 KFold 来实现。

7. from sklearn.externals import joblib

在模型训练完成之后，我们要把他们保存下来，所以需要 sklearn 当中的 joblib 这个类。

8. from sklearn.utils import shuffle

shuffle 函数可以将矩阵按行随机打乱，借用此函数，对我们的训练集进行随机混合。

9. from keras.models import load_model

我们在保存完 lstm 之后需要在测试以及应用的时候将其导入，load_model 的作用就是将之前训练好的模型导入使用。

2.3 数据收集模块

情报收集模块通过网络爬虫技术收集各个可靠情报源的数据。首先爬取Netlab上提供的总的黑名单域名作为黑名单情报库，再通过Alexa上提供的网站流量信息截取前100万的域名作为白名单情报库。然后继续利用爬虫技术在360、卡巴斯基等等国内外的可靠信息来源根据网站更新情况实时爬取恶意情报，提取出其中的域名加入到黑名单情报库中。爬取数据的过程中，先利用requests库获取网页内容并根据网页的编码方式读取，然后根据网页内容格式的不同利用不同的方法提取出相应的信息，或使用python原有的split等函数，或使用正则表达式提取，将文本中的域名，种类，时间等信息提取出来。最后使用tldextract库对提取的域名进行简化，提取出有用的字段。最后将得到的信息以域名，种类，时间的格式储存在csv文件中以进行情报的分析。除了开始收集到的大量数据之外，还每天分时段对各个情报网站进行数据爬取，实现情报库的不断扩充。

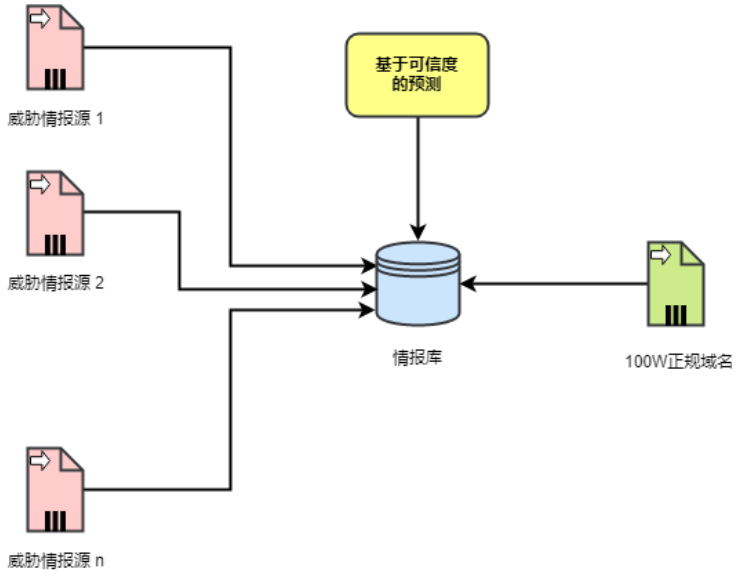

图 2-2 数据收集模块设计图

2.4 网络行为收集模块

网络行为收集模块以带有多种协议分析功能、能在各种大型站点运行的zeek/bro作为基础搭建。本模块通过对用户网络行为中IP、Domain等信息的分析，形成日志文件，实现对用户行为的实时监测。同时日志文件连接存储模块，以备后期处理。

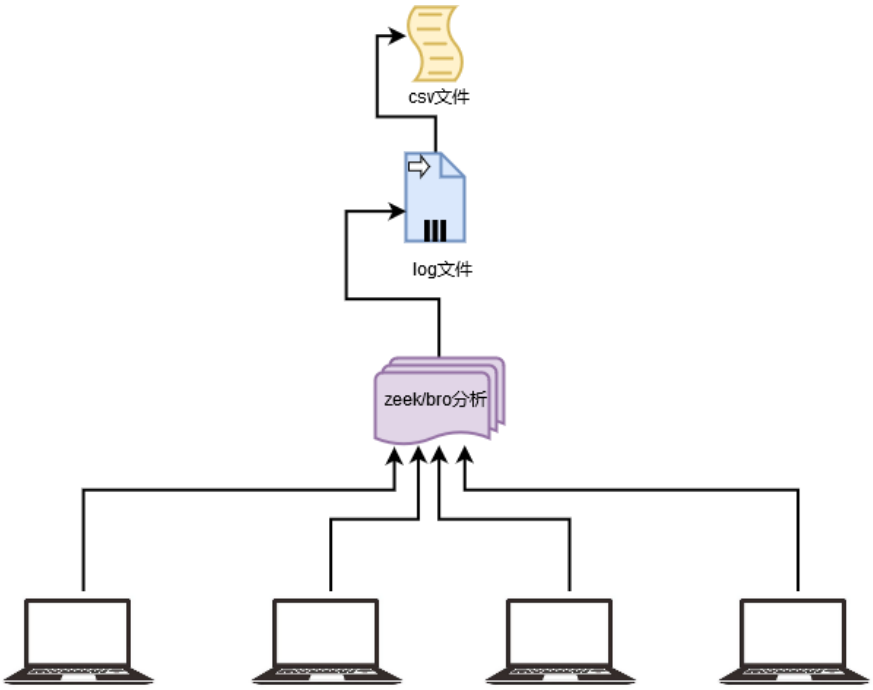

图 2-3 网络行为收集模块设计图

网络行为收集模块我们采用的是一款被动的流量分析器 Bro，利用其对主要的链路上所有深层次的流量进行大范围的监控。我们利用 libpcap 库来获取网络数据包，使我们可以忽略具体的链路层细节，从而获得较大的可移植性。libpcap 从上层的事件生成引擎获得的 Tcpdump 格式的网络流量过滤器，从搭载机器的网卡上过滤获取我们需要的网络流量，比如在对应设置中“tcp port 80 or port 23”的语句，会将从网络上获取到的流量过滤成源端口为 23 或 80 的 TCP 数据包，也就是对应分析目标的 HTTP 和 TELNET 协议相关的数据包。我们同时也搭载了 GeoIP 库，使得在检测出流量 IP 后我们可以获得其 IP 地址的具体地理位置，从而增加我们的数据形式和多样性。

利用这一模块，我们实时捕获网络中传输的数据包，获取到日志文件的扩展集。这些日志文件不仅全方位记录了所有线路上可见的每个链接，还记录了应用层传输，例如 HTTP 会话以及请求的 URL、服务器反馈、DNS 请求及反应、SSL 证书等内容。到此数据收集模块收集完毕，我们会对其日志文件进行提取，并且按照对应格式进行筛选存储，发至下一数据分析模块，对数据进行进一步分析。

模块的具体执行过程如下：

1. 调用函数 configure()来对当前执行环境进行检测，检测 Libpcap，Cmake 等必要依赖环境是否成功搭载；

2. 对 node.cfg 进行修改，设置支持当前搭载设备的监听端口；对 network.cfg 进行配置，针对是否存在多站点部署情况添加监控网段；修改 Broctl.cfg，设置日志文件生成频率，归档的日志文件以日期时间命名；

3. 用 root 身份运行流量分析器获取所需要的日志文件。一般日志文件使用 tab 分隔列，以可读的 ASCII 格式组织和保存。日志文件保存在设定的对应目录下，一般是 logs/current/。日志保存内容一般设置为：时间戳、连接的 UID，源地址和端口，目的地址和端口，协议细节信息。日志文件按照日期归档打包，名称分别根据其存储的日志内容来进行设定。例如：dns.log、http.log 等。

2.5 数据分析模块

在生活中我们常见的域名当中，几乎每个正规域名都有其明显的语义特征来方便人们的使用，而对于恶意域名，其具有较强的随机性，根据二者不同的特性我们设计并尝试了 12 种特征值来刻画出正规域名以及恶意域名的特点。12 个特征值如图所示，分别为：有意义单词占比、n 元模型的 1-gram、2-gram、3-gram、4-gram 和 5-gram、

数字占比、不同字母占比、不同数字占比、长度、元音字母占比还有字母数字的交换次数。在这个模块中，我们将对数据收集模块收集而成的威胁情报库中的白名单和黑名单分别进行特征值的提取，以便用于后续的机器学习模块。

序号	特征值名	序号	特征值名
1	有意义单词占比	7	数字占比
2	1-gram	8	不同字母占比
3	1-gram	9	不同数字占比
4	3-gram	10	长度
5	4-gram	11	元音字母占比
6	5-gram	12	数字字母交换次数

表 2-1 域名分析得到的特征值

首先，我们在网上爬虫下来的黑名单和白名单均用 csv 格式文件存储。

之后，我们先在 linguistic_classifier.py 这个文件中对域名分析出前 6 个特征值，再在 new_train_h/bmd.py 中分析出剩下的 6 个特征值。其中，分析前 6 个时，我们采用 Phoenix:DGA-based Botnet Tracking and Intelligence 论文中提供的数据集 10000 个常用单词作为库，该库如下：

图 2-4 常用词库

对域名进行第一部分处理时，我们逐行读取这 10000 个单词，以 million 为例在 n 元模型中的 1-gram 将 million 分为 m、i、l、l、i、o、n 的 7 个字段，对这 7 个字母对应字典中的值进行统计操作，而 2-gram 将会以 mi、il、ll、li、io、on 的字段对域名进行上述统计，以此类推至 5-gram；统计完成之后，利用 n-gram 正则化的分数公式：$score=\frac{\sum_i number[field_i]}{the\ number\ of\ \ fields-n+1}$（其中 $number[field_i]$ 是第 i 个字段的统计次数）计算出正则化分数，从而完成第一部分的特征值计算。

在第二部分的处理中，我们仅对每个域名的每个字符进行逐一访问，统计出数字、不同数字、不同字母、元音等等的数据，最后给出比例以及域名的长度。

在两个部分均处理完后，我们将所有结果写入一个 csv 格式的文件当中，作为后续机器学习模块进行学习的特征值矩阵。

2.6 机器学习模块

机器学习模块将黑名单、白名单分析后得到的相关特征值以及黑白标签（黑名单标签为 1，白名单标签为 0），统一格式化处理后，得到训练集。通过 lstm、xgboost、随机森林、逻辑回归、朴素贝叶斯、KNN 等模型对于训练集进行训练，得到多个机器学习模型，运用 IVAPDD 对这些模型进行打分。选出可信度最高的 k 个模型，对这些模型进行融合，得到最终模型。

对于朴素贝叶斯、逻辑回归这些 sklearn 当中有的模型，我们直接调用 fit 函数对我们的数据集进行学习，不需要手动调参，但 xgboost 和 lstm 就需要我们调用 xgboost 和 keras 当中的相关函数，并不断调参，得到较好的模型。

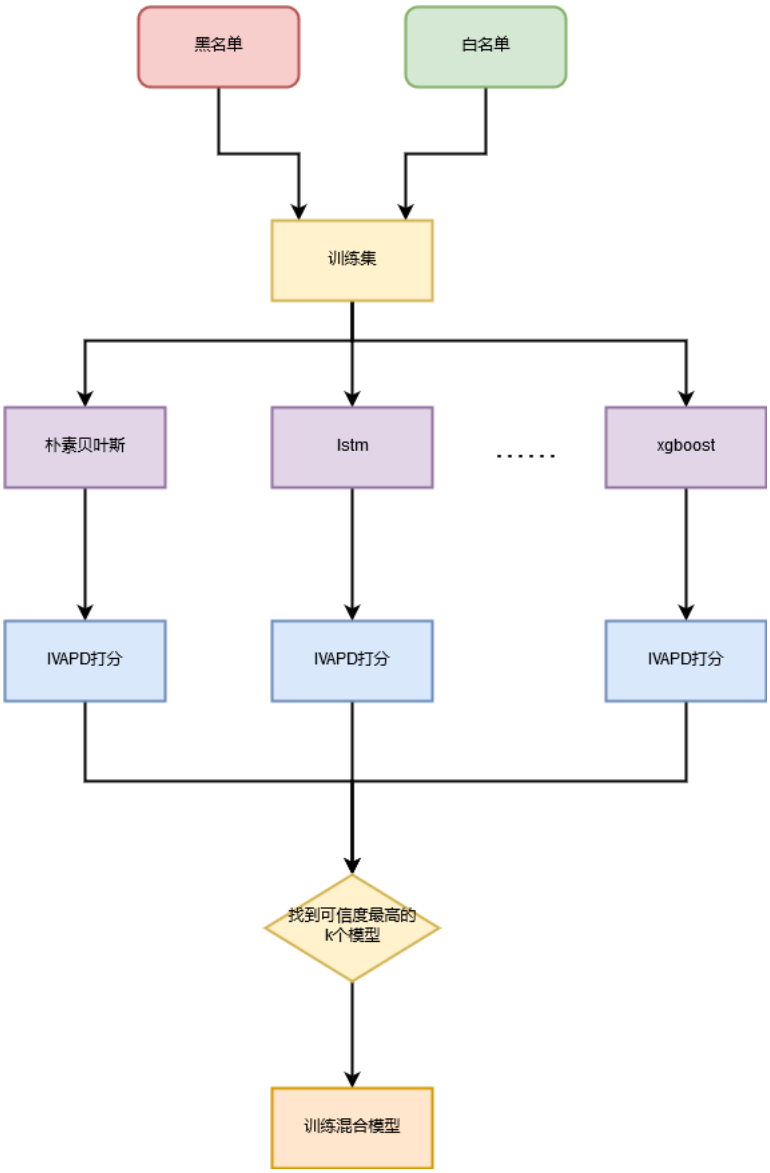

图 2-5 机器学习模块设计图

在多模型学习当中除了传统的随机森林、K近邻、朴素贝叶斯、bagging、逻辑回归、adaBoost、GradientBoosting以外，我们还采用了lstm和xgboost。

XGBoost，全称Xtreme Gradient Boosting，是以GBDT为基础的boosting 迭代算法、树类算法，主要应用于分类和回归，具有速度快，效果好，可以并行计算处理大量数据，支持自定义损失函数，可以进行正则化，具有高度灵活性，能够进行缺失值处理和剪枝，可以进行内置交叉验证，并可以在已有模型基础上继续训练等等优点，但是xgboost的发布时间短，工业领域应用较少，所以需要检验。

长短时记忆网络(Long Short Term Memory Network, LSTM)，是一种改进之后的循

环神经网络，规避了标准RNN中梯度爆炸和梯度消失的问题，可以解决RNN无法处理长距离的依赖的问题，学习速度更快，目前比较流行。

首先我们需要对收集来的黑白名单进行整合并随机打乱，得到训练集，其中用到了DataFrame和csv当中的一些操作函数。

```
def update_old():
    df=pd.DataFrame
    basepath=os.path.abspath(os.path.dirname(__file__))
    for info in os.listdir(basepath+"/old"):
        _info=info.strip('.csv')
        domain = os.path.abspath(basepath+"/old") #获取文件夹的路径
        info = os.path.join(domain,info) #将路径与文件名结合起来就是每个文件的完整路径
        data = pd.read_csv(info,sep='\t',header=None)
        if _info=="result_hmd(1970)":
            df=data
        else:
            df=pd.concat([df,data])
        print (data)

    data = pd.read_csv(basepath+'/result_bmd.csv',sep='\t',header=None)
    print (data)
    df=pd.concat([df,data.iloc[10000:]])
    df=df.sample(frac = 1)
    temp=df.iloc[:]
    print(temp)
    temp.to_csv("train_data.csv")
```

图 2-6 更新trail代码

对于sklearn当中的模型，我们只需要生成算法相应的clf类，并调用fit函数对训练集进行拟合，最后保存模型。以随机森林为例：

```
clf1 = RandomForestClassifier()
clf1.fit(train,label)
joblib.dump(clf1, "rfc.model")
```

图 2-7 随机森林模型训练代码

而对于xgboost，我们需要将其中的类和函数包装一下以供我们使用，我们需要得到类XGBC，其中有__init__、fit、predict_proba三个函数。__init__函数当中设定了一些需要调节的参数（通过后续调参获得较好的参数），fit函数对训练集进行拟合，predict_proba由给定的features得到预测结果。

```
class XGBC(object):
    def __init__(self, num_round = 2, max_depth = 2, eta= 1.0, min_child_weight = 2, colsample_bytree = 1, objective = 'multi:softprob'):
        self.max_depth = max_depth
        self.eta = eta
        self.colsample_bytree = colsample_bytree
        self.num_round = num_round
        self.min_child_weight = min_child_weight
        self.objective = objective
    def fit(self, train, label):
        dtrain = xgb.DMatrix(train, label = label, missing = -999)
        param = {'max_depth':self.max_depth, 'eta':self.eta, 'silent':1,
                 'colsample_bytree': self.colsample_bytree, 'min_child_weight': self.min_child_weight, 'objective':self.objective,
                 'num_class':2}
        self.bst = xgb.train(param, dtrain, self.num_round)
    def predict_proba(self, test):
        dtest = xgb.DMatrix(test, missing = -999)
        ypred = self.bst.predict(dtest)
        return ypred
```

图 2-8 xgboost模型包装代码

为了构建lstm的模型，我们需要用到keras当中的Sequential类，调用add、compile和fit等函数，去对我们的数据集进行拟合，最后用save函数来保存模型。

```
def lstm(features,labels):
    (x_train, y_train), (x_test, y_test) = split_data(features,labels)
    model = Sequential()
    model.add(LSTM(20,
                   input_shape=(lahead, 1),
                   batch_size=batch_size,
                   stateful=False))
    model.add(Dense(1))
    model.compile(loss='mse', optimizer='adam')
    print('Training')
    for i in range(epochs):
        print('Epoch', i + 1, '/', epochs)
        model.fit(x_train,
                  y_train,
                  batch_size=batch_size,
                  epochs=1,
                  verbose=1,
                  shuffle=False)
        model.reset_states()
    model.save('my_model.h5')
```

图 2-9 lstm模型训练代码

IVAPDD是由Venn-Abers发展而来的。Venn-Abers预测器提供了一种使用评分分类器作为基础ML方法来建立具有所需属性的分类的方法，而IVAPD的优点是，它在比其他方法更少限制的假设下实现校准。

IVAPD是近些年才提出的，所以没有相应的类库，我们需要自己根据相关论文写出来，其核心算法的伪代码如下：

Algorithm 1 Inductive Venn-Abers Predictive Distribution

INPUT: proper training set $T_P = \{(x_{-1}, y_{-1}), \ldots, (x_{-r}, y_{-r})\}$.
INPUT: calibration set $T_C = \{(x_1, y_1), \ldots, (x_h, y_h)\}$.
INPUT: testing example x_{h+1}.
INPUT: underlying predictor $P : (x, T) \to s$
for $i := 1, \ldots, r$ **do**
  $s_{-i} := P(x_i, T \setminus \{(x_{-i}, y_{-i})\})$
end for
find $(g_{-1}, \ldots, g_{-r})$ s.t. $\sum_{i=1}^{r}(g_{-i} - y_{-i})^2 \to \min$ wrt. $(s_{-i} \le s_{-j}) \Rightarrow (g_{-i} \le g_{-j})$
for $i := 1, \ldots, h+1$ **do**
  $s_i := P(x_i, T_P)$
  find s_{-j} which is the closest to s_i (may be not unique)
  $g_i := g_{-j}$ (take average if not unique)
end for
let $A := \{i = 1, \ldots, h : g_i = g_{h+1}\}$
let $\hat{Y} := \{y_i : i \in A\}$
OUTPUT:

$$\hat{P}_0\{y_{h+1} \le t\} := \frac{|\{\hat{y} \in \hat{Y} : \hat{y} \le t\}|}{|A| + 1}$$

$$\hat{P}_1\{y_{h+1} \le t\} := \frac{|\{\hat{y} \in \hat{Y} : \hat{y} \le t\}| + 1}{|A| + 1}$$

图 2-10 IVAPD算法伪代码

IVAPD能够给出基础ML算法的可信度，我们要选择可信度最高的k个的模型来构成我们的多模型。在多模型融合当中，我们设定了两个参数k和rate两个参数，分别为参与投票的模型个数和通过率，选用可信度最高的k个模型单独对威胁性未知的情报进行预测，如果认定该情报是威胁情报的模型的比率大于rate，就将其划定为威胁情报，否则将其划定为安全的。

2.7 情报繁殖模块

由于大多数恶意域名的生存时间极短，所以被动地收集各种情报源的威胁情并不能跟上威胁情报产生的速度。

在情报繁殖模块中，我们把通过网络行为收集模块获取到的大量用户的域名访问记录用现有的机器学习模型进行预测，然后选择其中可信度较高的恶意域名加入到我们的威胁情报库中。最后我们用更新后的训练集重新训练我们的机器学习模型。

重新训练的机器学习模型由于使用了最新的数据作为训练集，所以对于当下的威胁情报有更好的响应，准确率和召回率都有所提升，进而能够繁殖出更多的威胁情报，形成一个良心循环。极大地提高了本系统对于不断更新迭代的威胁情报的适应程度。

2.7 数据存储模块

在数据存储模块，我们使用CSV文件格式统一储存，来自情报收集模块的域名数据、数据分析模块的特征值信息，以及机器学习结果和统计学习结果，并最终将所有需要可视化展示的数据的 CSV 文件，上传至MySQL数据库，最后进行可视化展示。

储存的数据包括情报收集模块中收集的威胁情报（黑名单）的域名，种类和出现时间，正规域名（白名单）的域名，数据处理模块的网络行为分析数据以及对情报库和 zeek/bro 提取域名的特征值，数据分析模块的机器学习结果（域名得分）以及统计学习的结果。将上述数据将上传到可视化平台，在可视化平台中展示出本系统对数据的分析结果。

CSV文件以纯文本形式存储表格数据，非常适合本系统中对于情报库，网络行为分析数据、域名特征值，域名得分和IVAPD打分结果等数据的存储，并且本系统主要基于 python 开发，python有专门的的pandas库支持对CSV文件的快速写入和读取。其次，针对情报库、特征值、机器学习打分结果和统计学习结果可直接存储为 CSV 文件格式，而针对网络情报分析模块，在浏览网页时，zeek/bro 会首先生成 log 文件，通过 extract_log.py 文件转换成 CSV 文件格式。

统一的数据存储格式便于管理，同时也为可视化平台的数据导入奠定了基础。可视化平台嵌入了 MySQL 数据库，用于整合存储本系统上述模块的数据结果，并为可视化平台提供强大的大数据内部支持。

2.8 可视化模块

默认情况下，superset 是把元数据保存到 SQLite 中的，我们搭建了MySQL数据库，同时将MySQL数据库与superset进行连接，然后将情报 CSV 文件， 网络行为分析文件，特征值文件，P-Value 文件上传到平台，平台能够展示出对数据的分析情况，同时，可基于我们的MySQL数据库在平台上输入 SQL 语句对数据进行查询。我们运用此功能，将收集到的网站的相关信息生成图表展示给用户，使用户更加便捷地了解网站的信息。同时将本系统生成的结果展示给用户，让用户能够获得我们对该网站的判断。

我们可以将获取的网站整体信息以 table view、 sunburst、 partition diagram 、pie chart、bar chart的形式展示出来，同时可以将网站的某一具体信息(例如destination信息)以name cloud、 pivot table 、time series – line chart 、pie chart 的形式展示出来，将

我们使用的特征值信息等以Name Cloud (文字云图) 形式展示出来。最后，可以将这些图统一存储到一个Dashboard上，一起进行展示。

在可视化模块，我们使用 Superset 进行图形可视化。Superset 是一款开源的 OLAP 及数据可视化前端工具，是由知名在线房屋短租公司 Airbnb 公司开源的一款数据探 索及数据可视化工具，初始项目名称为 Panoramix 后改名 Caravel，近期项目改名为 Superset。Superset的数据库连接信息存储在元数据库，因此，他们使用了 cryptography 密码库来对连接信息进行加密，而且我们需要在 virtualenv 虚拟环境中安装 superset。

Superset的功能十分强大，主要有以下几点:

（1）支持的数据源:

目前原生支持的数据源有:MySQL、Postgres、Presto、Oracle、SQLite、Redshift、MSSQL 以及 Druid。对于前面的关系型数据库，Superset 通过将界面的操作转换成 SQL语句，提交给SQLAlchemy适配数据源查询并返回结果，对于Druid，Superset将 界面的操作转换成 Druid 的 API 进行查询并返回结果，与 Druid 的深度集成，可以实 现大规模海量数据的 OLAP 分析和实时探索。

其中，Druid 是一个基于分布式的快速列式存储，也是一个为 BI 设计的开源数据 存储查询工具。Druid 提供了一种实时数据低延迟的插入、灵活的数据探索和快速数 据聚合。现有的 Druid 已经可以支持扩展到 TB 级别的事件和 PB 级的数据了，Druid 是 BI 应用的最佳搭档。

（2）数据缓存:

我们可以查看进行查询表的 sql，也可以把查询导出为 json 或者 csv 文件。它有自己的 sql 编辑器，我们可以在里面来编写 sql。配置好了我们想要的图表之后我们可以把它添加到 dashboard 进行展示，为了提高并发查询下的性能，还支持数据缓存，可配置将数据缓存至 Redis、Memcache 或者本地文件系统，来加速dashboard的查询，不必要每次都去查询数据库。

（3）数据可视化:

Superset 支持十几种可视化图表，我们可以通过连接数据库，去对数据库中的单个表进行配置，展示出柱状图，折线图，饼图，气泡图，词汇云，数字，环状层次图，

有向图，蛇形图，地图，平行坐标，热力图，箱线图，树状图，热力图，水平图等图，我们可以操作视图，利用数据库创建视图，以及在 superset 中对表格进行修改和展示。

我们将数据生成 csv 文件并使用 MySQL 进行存储，使用 superset 平台对 MySQL 的数据库中存储的有关网站的信息进行展示，例如将我们收集到的网站的信息以Pivot table或table view的方式显示出来给用户，将每个网站的具体某个信息的相关内容以饼图、柱状图等方式进行显示，并生成 Dashboard，给予用户更加直观的展示，用户可以在dashboard中很直观地大概了解网站相关信息，也可以点击相应表格进行详细的了解。

2.9 本章小结

本章首先从整体对系统进行了初步的介绍，展现了系统的整体框架和功能结构，明确了技术路线。接着，从各个模块对系统进行了进一步的阐述与设计，详细说明了数据收集、数据分析、网络行为收集、机器学习、模型融合、情报繁殖、数据存储、可视化模块的设计思路和功能，以及互相之间的联系。

第三章 作品测试与分析

3.1 测试方案

针对系统的情报收集、网络数据分析、数据分析、机器学习、情报繁殖、存储模块、可视化模块分别进行测试。

3.2 测试设备

处理器：Inter(R)Core(TM)i7-7500U CPU @ 2.70GHz 2.90GHz

操作系统：Windows10 64 位

内存：4GB

硬盘：80GB

环境语言：python 3.7

网络适配器：NAT

设备名称：DESKTOP-D470HST

3.3 数据收集模块

首先从 Netlab 上爬取 100 万的基础黑名单情报库，使用正则表达式或者 split 函数获取数据的域名，family 和出现时间。再用 tldextract 库对得到的域名进行加工，去除无效的部分，保留有用的部分。

pugqdedp.org	conficker dga (malware)	2019/5/17
xzmtrmk.info	conficker dga (malware)	2019/5/17
jhvbt.com	conficker dga (malware)	2019/5/17
ltpyqth.net	conficker dga (malware)	2019/5/17
uogndqcw.info	conficker dga (malware)	2019/5/17
vldknqsqfgr.com	conficker dga (malware)	2019/5/17
tuujbh.info	conficker dga (malware)	2019/5/17
kuqkt.info	conficker dga (malware)	2019/5/17
qgvfusr.org	conficker dga (malware)	2019/5/17
lpjympozb.net	conficker dga (malware)	2019/5/17
cxgcqfvd.com	conficker dga (malware)	2019/5/17
cramhocrr.biz	conficker dga (malware)	2019/5/17
oldeimijy.net	conficker dga (malware)	2019/5/17
vqauracchd.biz	conficker dga (malware)	2019/5/17
lxofebfk.biz	conficker dga (malware)	2019/5/17
gvuytcakkr.org	conficker dga (malware)	2019/5/17
wullb.net	conficker dga (malware)	2019/5/17
savjqlalo.net	conficker dga (malware)	2019/5/17
gshuacpy.com	conficker dga (malware)	2019/5/17
wwdeyd.com	conficker dga (malware)	2019/5/17
cnihjidup.net	conficker dga (malware)	2019/5/17

表 3-1 数据储存格式

wvbxdm.com
jjnqfog.biz
nrlaoqjgqj.biz
onnwxlddcn.com
uybvwh.info
jojhrcx.com
zixaqgjsgdr.com
gepszovbf.com
extiftw.net
orineen.com
heyqvdojdk.com
kdddmvrk.net
jdrzdsciph.biz
ttujxceo.com
dyugbrg.net
bkbqvwbf.com
fnvqdorj.net
lmukghf.com
vgykxrk.net
syxftq.net
poawdgpxkgy.com
aaihjdhqh.biz

图 3-1 威胁域名

从 20 余个域名情报源中获取黑名单：

在从 20 余个域名更新情报库时，根据不同网站格式的不同，采用不同的正则表达式提取，每天定时运行爬虫程序，更新情报库。对于有时间标签的网页，通过比较时间标签和现在时间来判断是否爬取该网站的内容，对于没有时间标签的网页，则爬取到上次停止的位置。

```
r"\A127.0.0.1\s+(.+)\Z"
```

图 3-2 正则表达式 1

```
r"(?m)^([\w.]+)\s+2\d{3}\-"
```

图 3-3 正则表达式 2

3.4 网络行为收集模块

Bro 网络安全监控器，从 Bro 官网下载源码包，编译安装，修改环境变量，网址：https://www.Bro.org/；编译安装命令为：./configure Make Make install；将可执行文件路径添加到环境变量中：export PATH=/usr/local/Bro/bin:$PATH。

配置文件，打开/usr/local/Bro/etc/node.cfg 配置要监视的节点。测试中先通过 ifconfig -a 命令查看本机网络接口，并将 node.cfg 中的 interface 参数改为对应值 ens33。打开/usr/local/Bro/etc/networks.cfg 配置节点的专用网络。测试中先通过 ip addr show 检查服务器接口的网络地址，并修改对应的 Public IP space 和 Private IP space 参数。打开/usr/local/Bro/etc/Broctl.cfg 配置日志记录时间，修改 LogRotationInterval，令 Bro 记录间隔时间为 1min，实现较为及时的记录更新。

使用 BroControl 运行 Bro，执行命令：sudo /usr/local/Bro/bin/Broctl deploy，运行结果如图 3-4。

```
Welcome to BroControl 1.7-61

Type "help" for help.

[BroControl] > start
waiting for lock (owned by PID 2800) ...
Error: Unable to get lock
[BroControl] > install
removing old policies in /usr/local/bro/spool/installed-scripts-do-not-touch/sit
e ...
removing old policies in /usr/local/bro/spool/installed-scripts-do-not-touch/aut
o ...
creating policy directories ...
installing site policies ...
generating standalone-layout.bro ...
generating local-networks.bro ...
generating broctl-config.bro ...
generating broctl-config.sh ...
[BroControl] > start
starting bro ...
[BroControl] >
```

图 3-4 Bro 运行结果

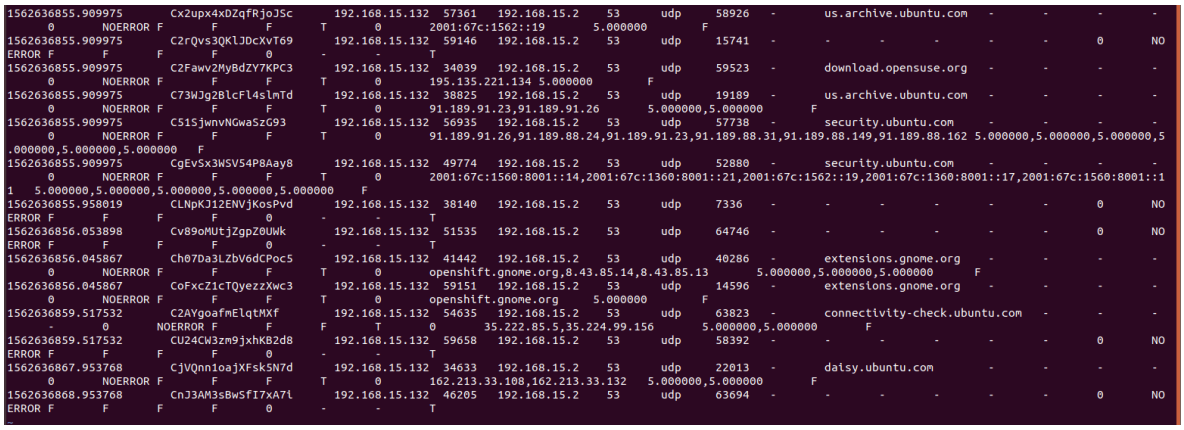

图 3-5 Bro 日志

访问网页并查看日志记录。

每分钟形成一个 dns 压缩文件，形成的文件如图 3-5。

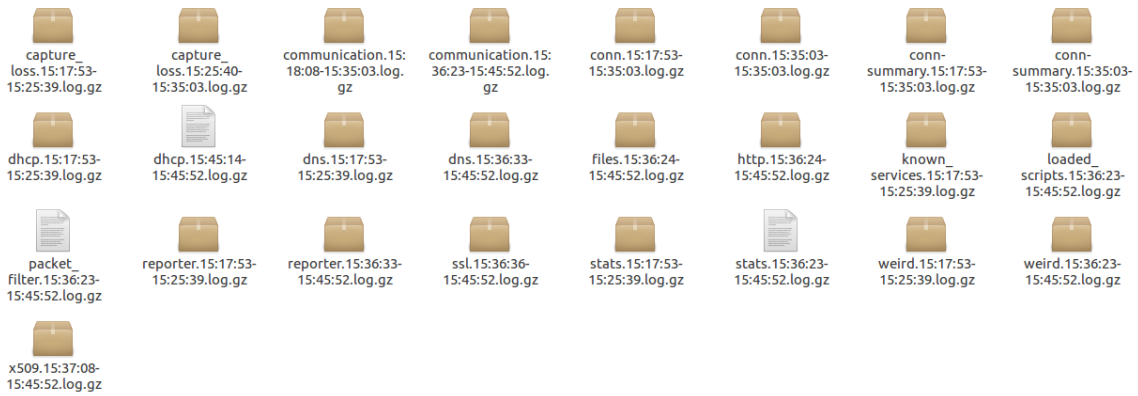

图 3-6 DNS 压缩文件

打开压缩文件，查看记录的访问网址，如图 3-6。

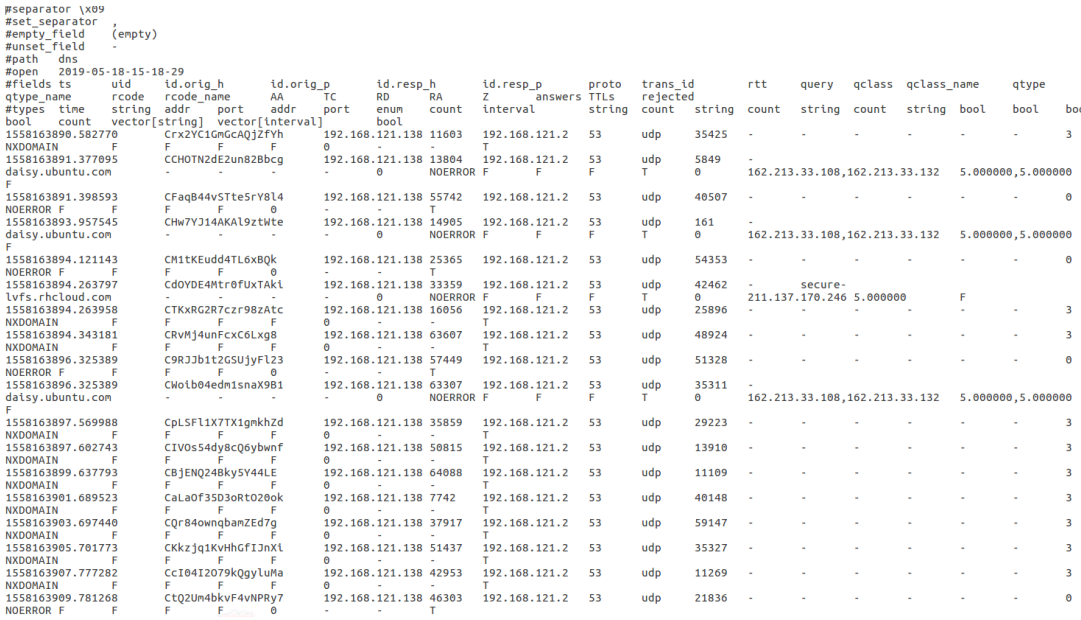

图 3-7 访问网页的记录信息

处理文件，提取访问信息，将日志记录转 csv 文件。

某段时间 bro 对网络行为检测的结果如下表所示：

Ports	ratio	Sources	ratio.1	Destinations	ratio.2	Services	ratio.3	Protocols	ratio.4	States	ratio.5
53	40.5	192.168.231.138	97.5	192.168.231.2	40.5	-	58.0	6	55.7	SHR	58.8
	43.4	192.168.121.138	69.8	192.168.121.2	44.0	dns	55.5	17	84.1	SHR	45.1
80	27.4	fe80::f99a:5b07:db96:bebc	0.9	122.205.109.48	4.0	dhcp	0.2	1	0.2	RSTRH	3.2
443	28.1	192.168.231.1	1.0	172.217.24.14	7.8	dns	41.9	17	44.2	OTH	33.6
	86.7	192.168.231.138	100.0	60.221.218.25	40.0	-	100.0	6	100.0	RSTRH	53.3
5353	13.2	fe80::a8e:2e59:ac83:bc3b	8.8	ff02::fb	6.6	dhcp	2.2	1	5.5	S0	17.6
5355	13.2	192.168.121.1	11.0	ff02::1:3	6.6	-	42.3	6	10.4	OTH	35.2

表 3-8 网络行为检测信息

将访问源 IP 进行可视化展示，如下图所示：

Sources	SUM(ratio.1)
192.168.231.138	198
192.168.121.138	69.8
192.168.121.1	11.0
fe80::a8e:2e59:ac83:bc3b	8.80
192.168.231.1	1.00
fe80::f99a:5b07:db96:bebc	900m

图 3-9 主机访问的源 IP

将其以饼状图更直观的展示出来：

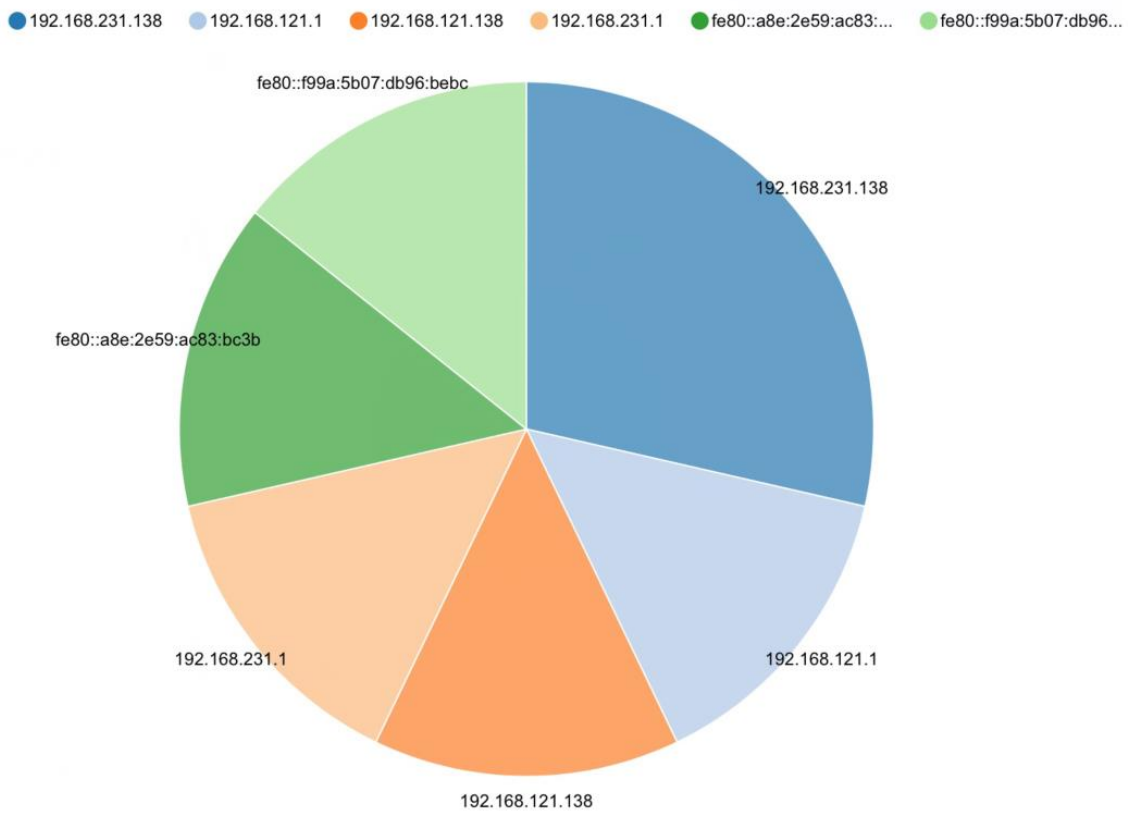

图 3-10 主机访问的源 IP 比例

将访问目的 IP 进行可视化展示：

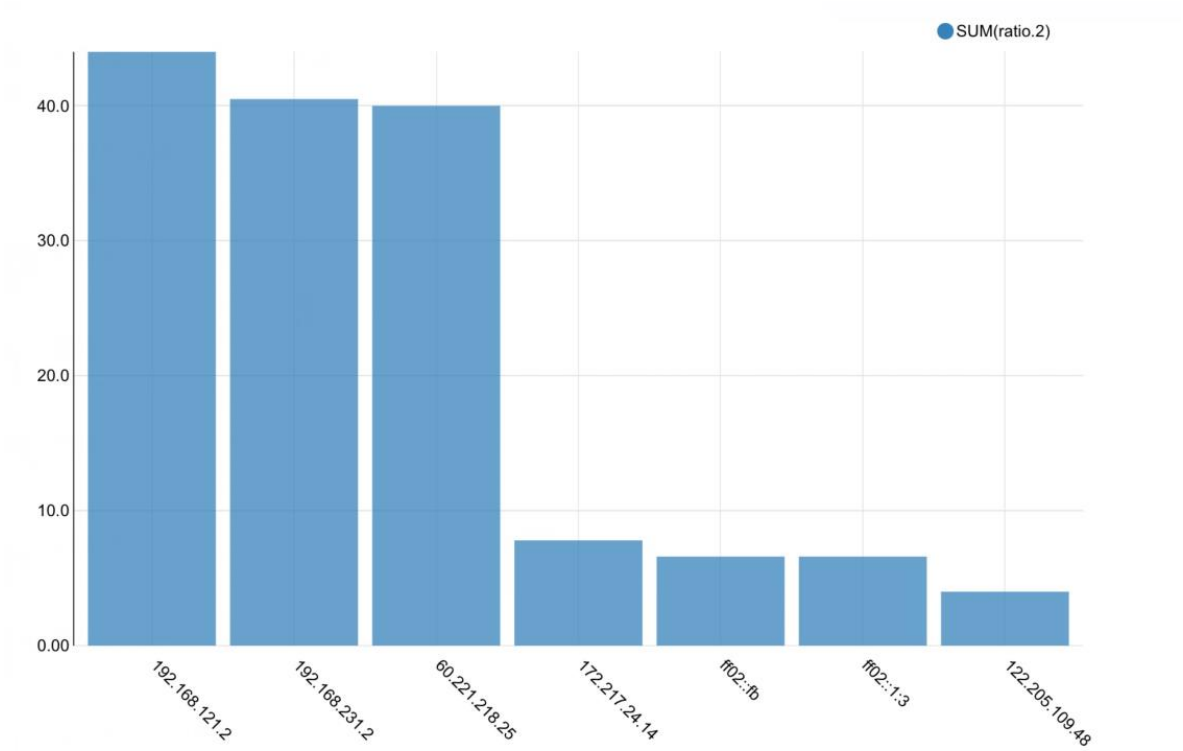

图 3-11 主机访问的目的 IP

将访问目的 IP 以饼状图直观的展示出来，以看到各个 IP 所占的比例：

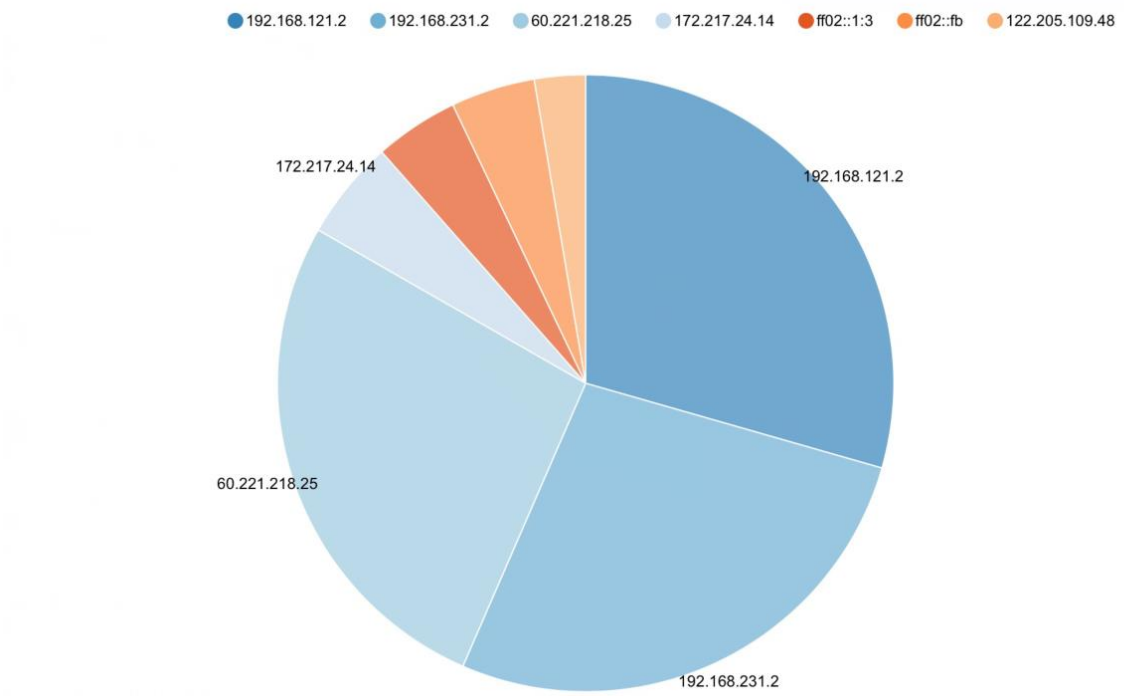

图 3-12 主机访问的目的 IP 的比例

访问的网站所占用的端口如下图所示：

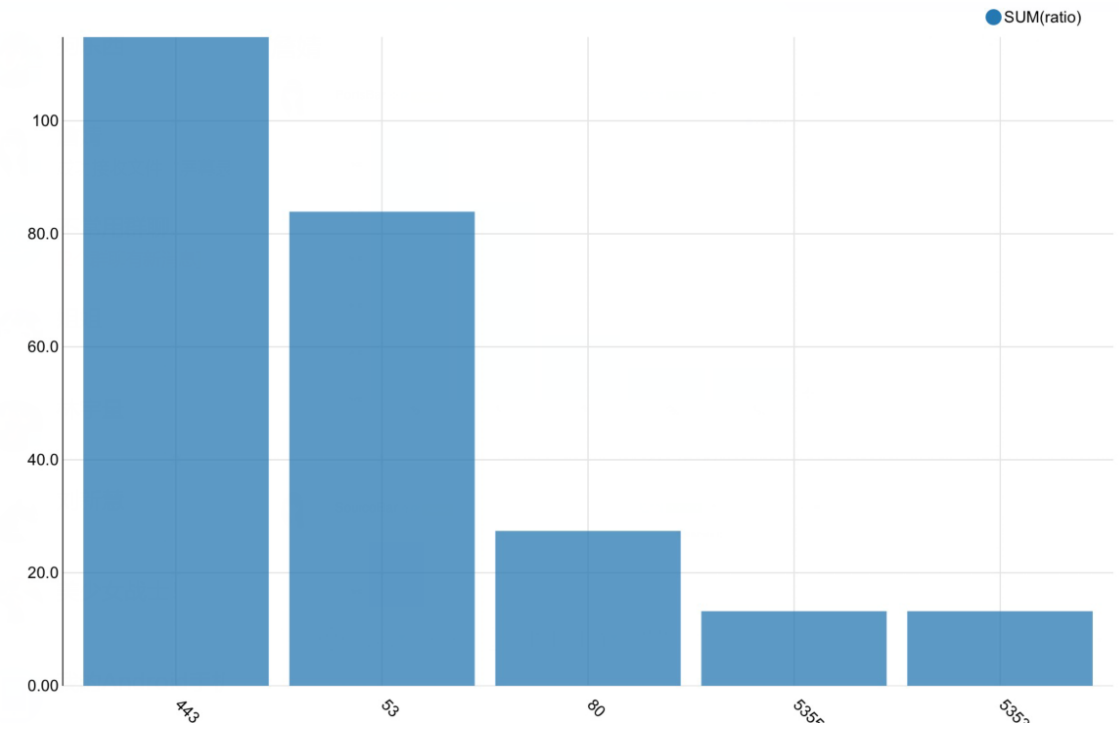

图 3-13 占用端口情况

各类访问状态所占的比例如下图所示：

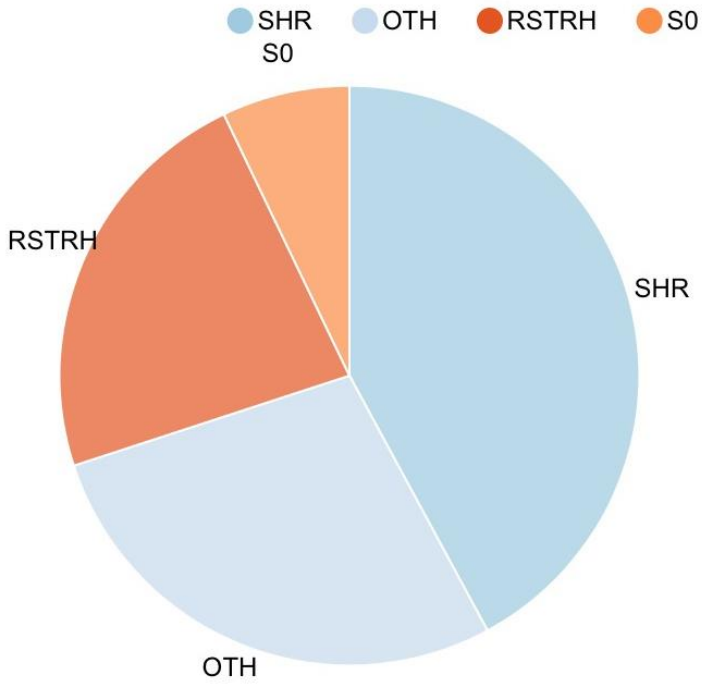

图 3-14 访问状态

3.5 数据分析模块

运行 linguistic_classifier.py 和 new_train_hmd.py 得到的一组数据（数据选择的是 2019 年 5 月 6 日爬虫得到的黑名单），如表所示为其中的一部分：

Meaningful words ratio	1-gram	2-gram	3-gram	4-gram	5-gram	Number ratio	Different letter ratio	Different number ratio	Length	aeiou ratio	Change times
0.6	2981.6	208.4444444	29.875	5	2.333333333	0.1	0.9	0.1	10	0.272727273	1
0	2720.125	112.7142857	3.666666667	0.6	0.5	0.125	0.75	0.125	8	0.333333333	1
0.923076923	4895.769231	591.75	65.18181818	11.1	2.777777778	0.076923077	0.615384615	0.076923077	13	0.357142857	1
0.833333333	4017	630.8	133	6	1.5	0.166666667	0.833333333	0.166666667	6	0.285714286	1
0	538.75	0	0	0	0	0.5	0.25	0.5	4	0	3
0.272727273	3791.818182	243.9	12.77777778	0.125	0	0	0.818181818	0	11	0.333333333	0
0.5	3461.083333	374.1818182	41	5.555555556	1.75	0.25	0.666666667	0.25	12	0.230769231	1
0.8	3211.6	342.8571429	39.53846154	4.25	1.363636364	0.133333333	0.6	0.133333333	15	0.25	1
0	2265.4	176.2222222	5.25	0	0	0.3	0.6	0.3	10	0.272727273	3
0.5	3377.666667	295.6	53.75	3.666666667	0	0.333333333	0.666666667	0.333333333	6	0.285714286	1
0.666666667	4042.5	173.8	15.75	1.666666667	0	0.166666667	0.833333333	0.166666667	6	0.428571429	2
0.714285714	3037.142857	226.8333333	17.2	4.25	1.333333333	0.142857143	0.714285714	0.142857143	7	0.25	2
0.571428571	2509.285714	89.16666667	4.2	0.5	0	0.285714286	0.714285714	0.285714286	7	0.25	3
0.857142857	4382.761905	342.25	93.05263158	44.77777778	15.05882353	0	0.619047619	0	21	0.409090909	0
0.5	4411.375	491.8571429	33	4.4	0.25	0	0.75	0	8	0.444444444	0
0.6	4313.2	208.25	14.33333333	0	0	0	0.8	0	5	0.333333333	0
0.833333333	3428.583333	275.5454545	19	1.555555556	0.75	0	0.833333333	0	12	0.230769231	0
0.3	3230.6	210.4444444	10.625	0	0	0	0.7	0	10	0.363636364	0

表 3-15 数据分析模块的到的特征值

运行 linguistic_classifier.py 和 new_train_bmd.py 得到的一组数据（数据选择的是 2019 年上半年爬虫得到的白名单），如表所示为其中的一部分：

Meaningful words ratio	1-gram	2-gram	3-gram	4-gram	5-gram	Number ratio	Different letter ratio	Different number ratio	Length	aeiou ratio	Change times
0.777777778	4185.666667	382.125	26.14285714	2.666666667	0	0	0.888888889	0	9	0.333333333	0
1	5403.7	417.8888889	32.5	2.142857143	0.333333333	0	0.7	0	10	0.4	0
0.857142857	4253.5	384.3846154	36.08333333	5.545454545	1.7	0	0.714285714	0	14	0.428571429	0
0.75	4555.375	279.4285714	21.16666667	1.2	0	0	0.75	0	8	0.5	0
0.272727273	4685.090909	301.6	19.11111111	1.875	0	0	0.727272727	0	11	0.454545455	0
0.7	4212.3	267.8888889	4.125	0.285714286	0	0	0.9	0	10	0.4	0
0.823529412	4472.882353	317.375	35	5.285714286	0.692307692	0	0.647058824	0	17	0.235294118	0
0	2671.333333	115.6	1.25	0	0	0	1	0	6	0.333333333	0
0.375	4765.625	322.7142857	17.33333333	0.2	0	0	0.625	0	8	0.375	0
0	859.25	0.333333333	0	0	0	0	1	0	4	0	0
0.5	4827	501	29	0	0	0	0.833333333	0	6	0.166666667	0
0	3897.25	227.3333333	6	0	0	0	0.75	0	4	0.5	0
0.416666667	3869.666667	200.9090909	24.3	2.888888889	1	0	0.75	0	12	0.416666667	0
0	4414.75	566	7	0	0	0.25	0.75	0.25	4	0.5	1
0.818181818	4079.636364	475.3	162.4444444	92.5	39.71428571	0	0.727272727	0	11	0.454545455	0
0.5	3260	250	20	1	0	0	1	0	6	0.333333333	0
0.363636364	3842.727273	269.9	10.22222222	0.125	0	0	0.818181818	0	11	0.363636364	0
1	4337.125	439.8571429	75.5	3.2	0	0	0.875	0	8	0.375	0

表 3-16 数据分析模块得到的白名单特征值

3.6 机器学习模块

3.6.1 环境搭建

随机森林、lstm、xgboost等模型的测试在python3.7版本下进行，需要预先使用pip命令安装numpy、pandas、sklearn、keras等python库。

3.6.2 测试数据

我们将2019年5月份收集来的较新的威胁情报以及白名单中的一小部分（不在训练及当中出现）随机打乱，作为测试集。

3.6.3 测试代码

我们首先导入之前训练好的模型，对测试数据进行预测，再将预测标签和真实标签作为参数，传递给IVAPD的评分函数，得到模型的可信度。以随机森林为例：

```
def test_rfc_vennabers():
    print("test_rfc_vennabers")
    clf = joblib.load("rfc.model")
    pred=clf.predict(features)
    l1=[]
    correct=0
    for i in range(labels.shape[0]):
        if abs(labels[i]-pred[i])<0.000000001:
            correct+=1
        temp=(labels[i],pred[i])
        l1.append(temp)
    l2=list(pred)
    p0,p1=ScoresToMultiProbs(l1,l2)
    print(correct/labels.shape[0])
    print(p0)
    print(p1)
    df=pd.DataFrame()
      df['label']=labels
    df['pred']=pred
    df.to_csv(basepath+'/pred_result/rfc_result.csv')
    return p0,p1
```

图 3-17 随机森林测试代码

3.6.4 测试结果以及分析

在xgboost模型当中，各个特征值在预测中起到的重要性：

label	SUM(importance)
字母数字的交换次数	2.51k
元音字母占比	22.5k
长度	41.0k
不同数字占比	2.45k
不同字母占比	3.98k
数字占比	4.31k
5-gram	5.36k
4-gram	7.66k
3-gram	12.4k
2-gram	15.3k
1-gram	17.2k
有意义单词占比	65.4k

图 3-18 各特征值重要性

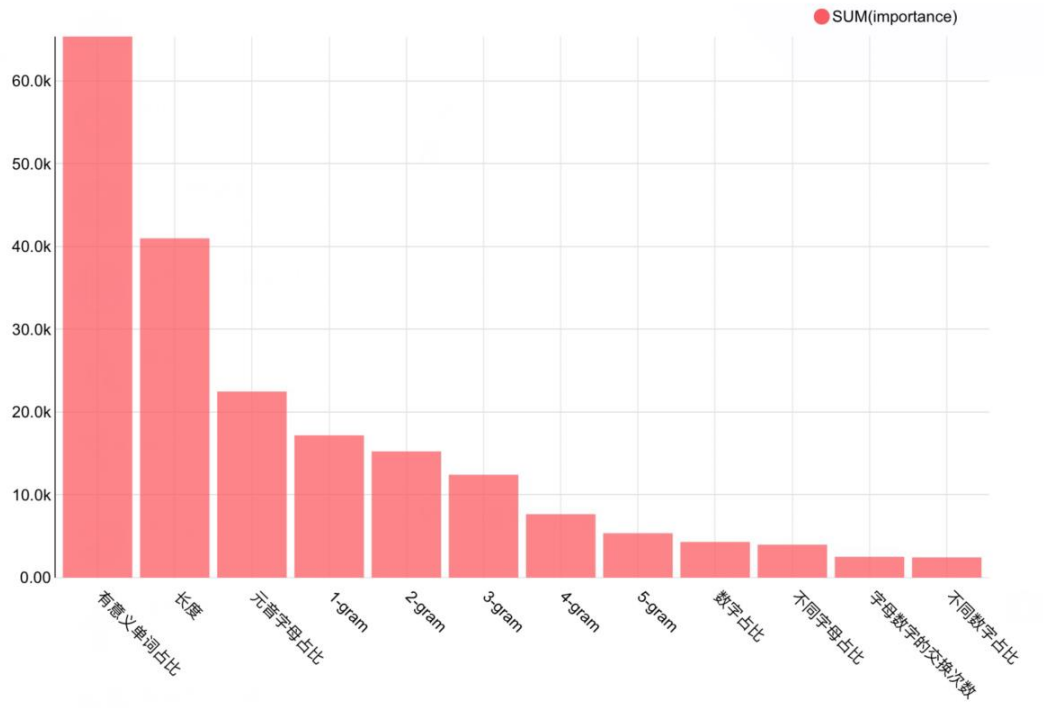

图 3-19 各特征值重要性柱状图

由上两张图可知，有意义单词占比在xgboost预测情报威胁性当中起着最重要

的作用，其次，元音字母占比、和域名长度也都站很重要的地位，再然后。，1-5gram在打分过程中占的重要程度也很高，5-gram到1-gram重要性递增。

将xgboost模型中特征值重要性以饼状图更直观的展示出来，如下图所示：

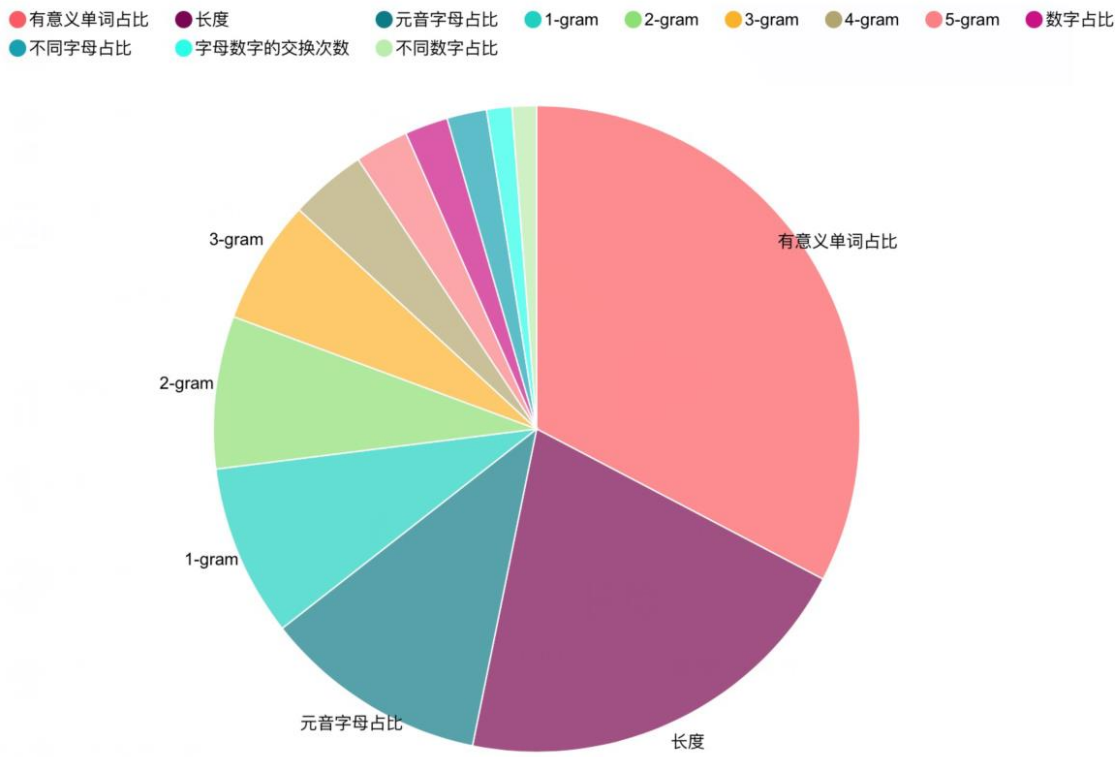

图 3-20 各特征值重要性饼状图

由饼图可以估计出，长度和有意义单词占比占了一半以上的重要性，1-5gram这5个特征值的占比总和占了全部的四分之一，而字母数字的交换次数、不同数字占比、不同字母占比和数字占比的作用非常的小。

将特征值重要性用词云展

示：

图 3-11 特征值重要性词云

不同的模型的可信度随时间的变化：

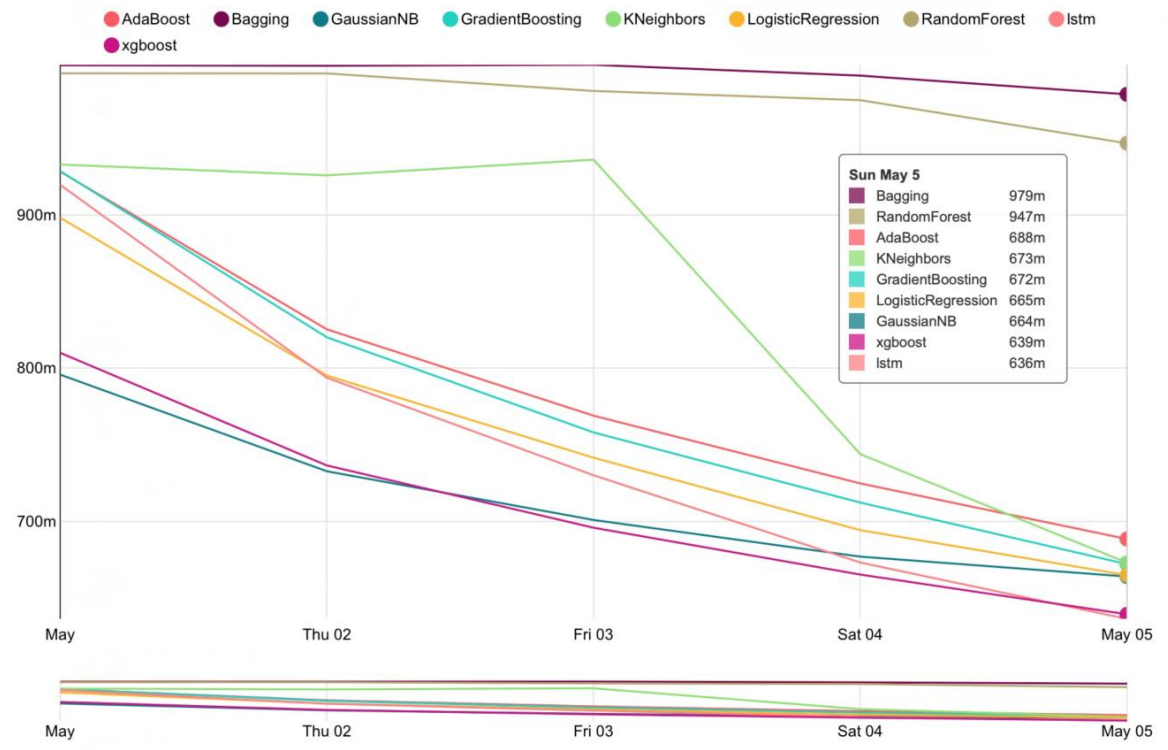

图 3-21 模型可信度随时间变化图

由上图可以看到，同样的训练集训练出来的模型，对于威胁情报的预测的可信度会随着时间的推移而逐渐降低，但降低的程度跟机器学习模型本身有很大的关联。Bagging和随机森林的可信度最高，而且随时间的变化比较小，在我们的多

模型当中，应该更加优先选择。由图可知，k近邻、逻辑回归、lstm、xgboost和朴素贝叶斯等模型的可信度随时间的推移下降的比较快，所以对于这些模型，我们一定要及时更新，防止模型可信度降低，模型失效。

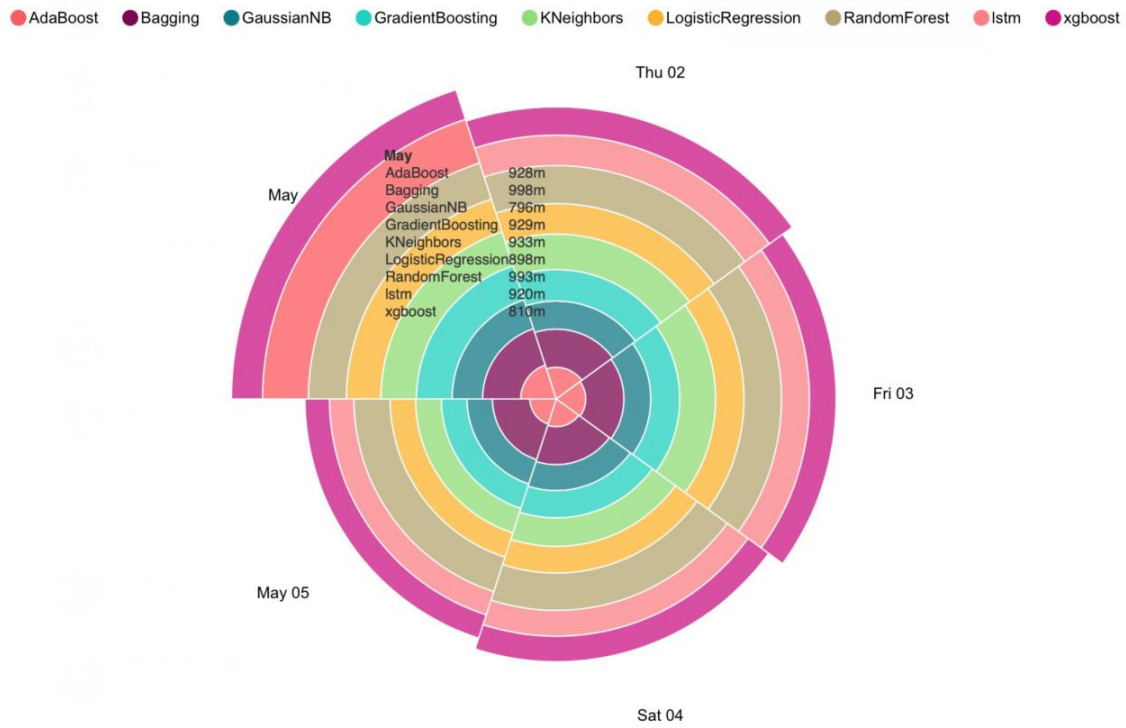

图 3-22 模型可信度

这张图可以看到各个模型的可信度都会随着时间的推移而降低，但是可信度较高的模型，即使在一段时间后，可信度还是领先于其他的模型，所以选取的模型一般不会根据时间而改变。

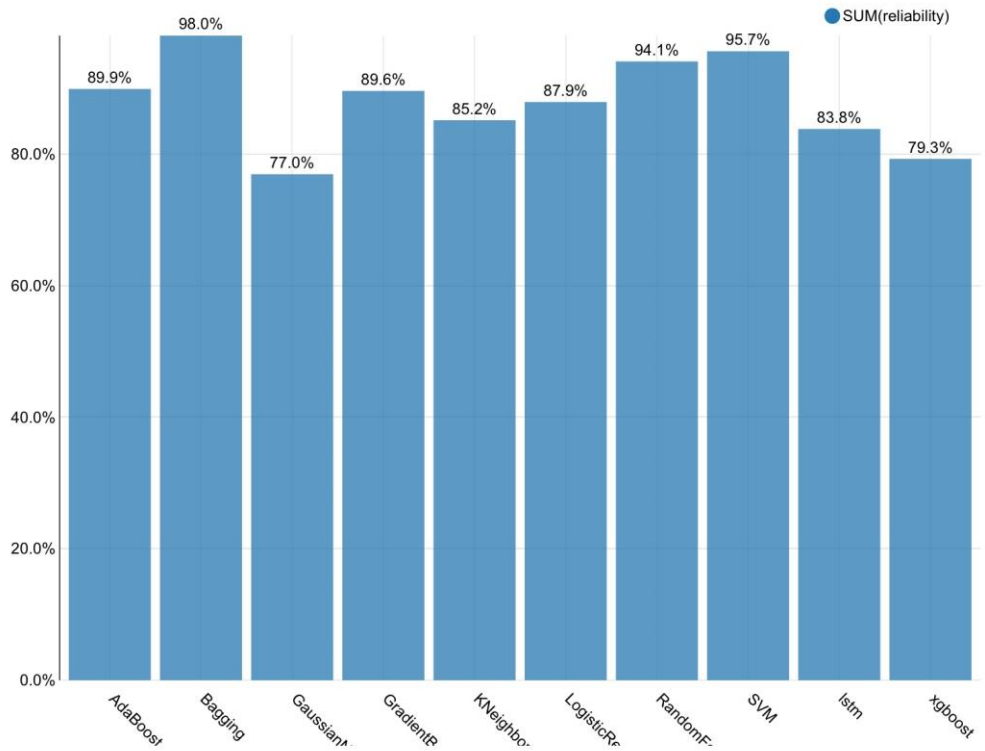

图3-23 模型可信度对比

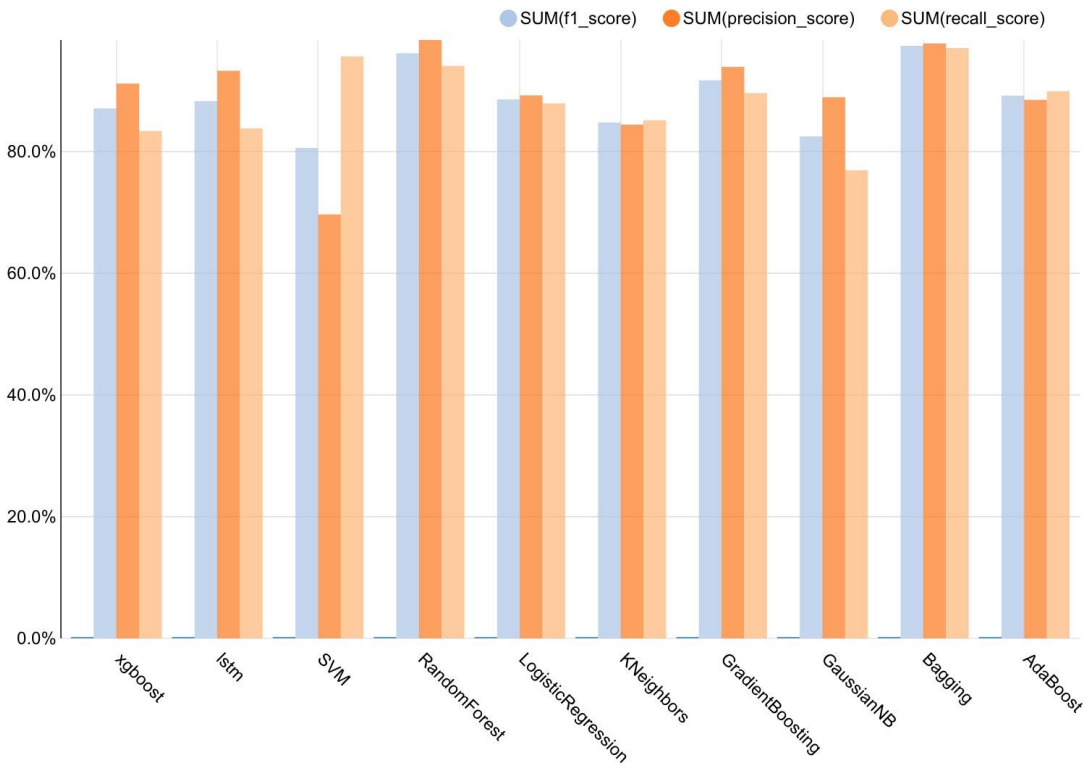

图3-24 单个模型准确率、召回率、F值

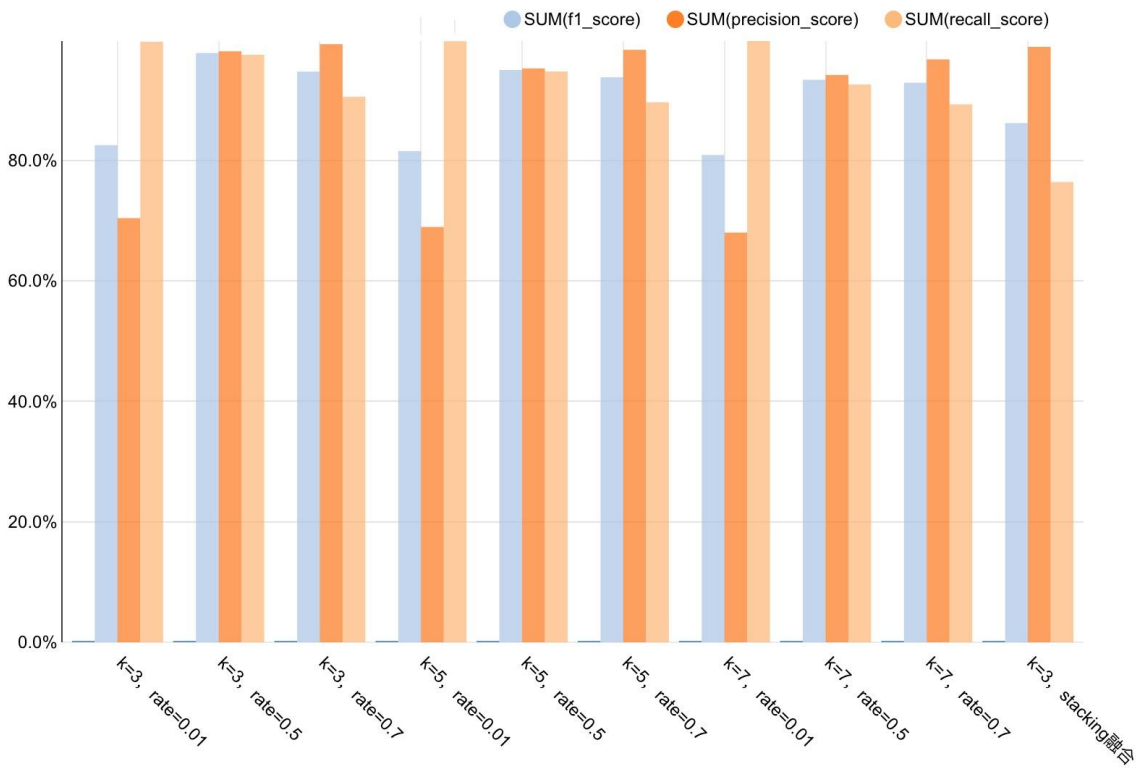

图3-25 融合模型准确率、召回率、F值

由上图可以直观的看出，各个融合模型均在准确率或召回率上有显著的提高。经过分析可以看出：多模型混合的准确率与k成反相关，与rate成正相关；而召回率相反，与k成正相关，与rate成反相关。分析各种不同的参数组合，可以发现rate=0.01的过滤法的召回率最高，能查出99.7%以上的威胁情报，而rate=0.7的stacking法的准确率最高，在k=3时达到了0.993。而相对而言，投票法在召回率和准确率上都不是最高，但综合起来是最好的算法，当k=3，rate=0.5时，f1_score得到了最高的0.9784，而且在准确率和召回率上都超过了表现最好的单一模型bagging。由此可见基于可信度的多模型融合方法的优越性。

3.7 情报繁殖模块

图 bagging模型可信度

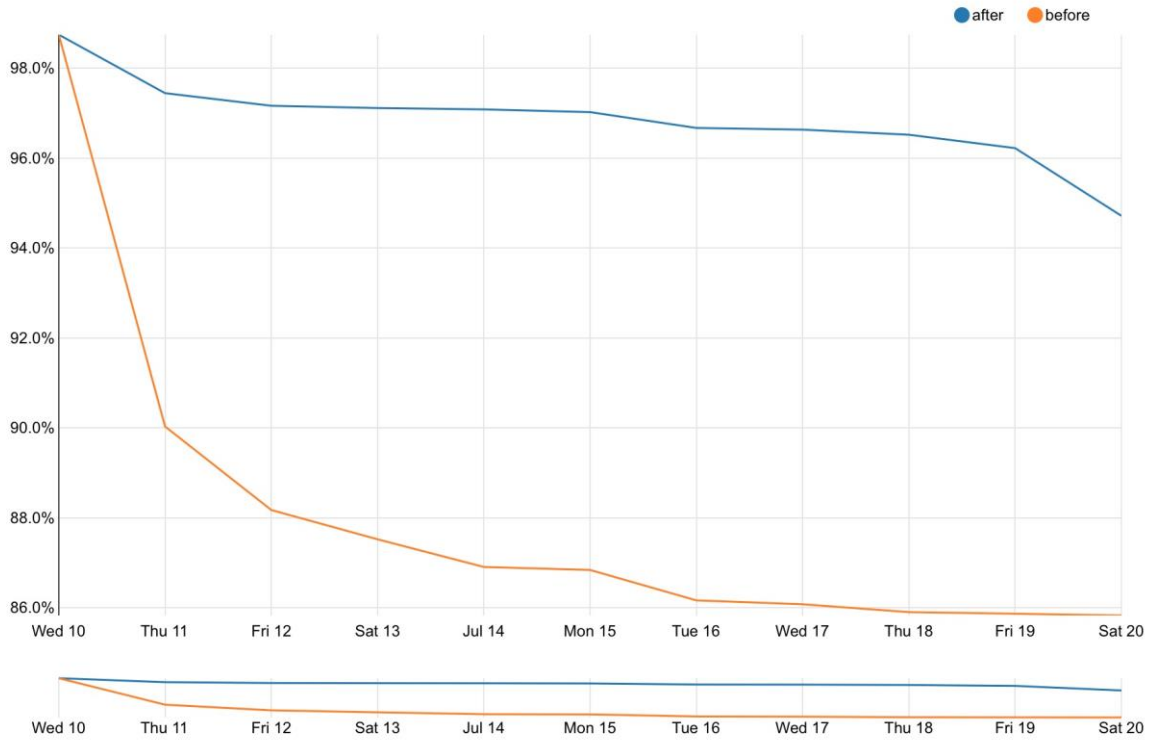

图 随机森林模型可信度

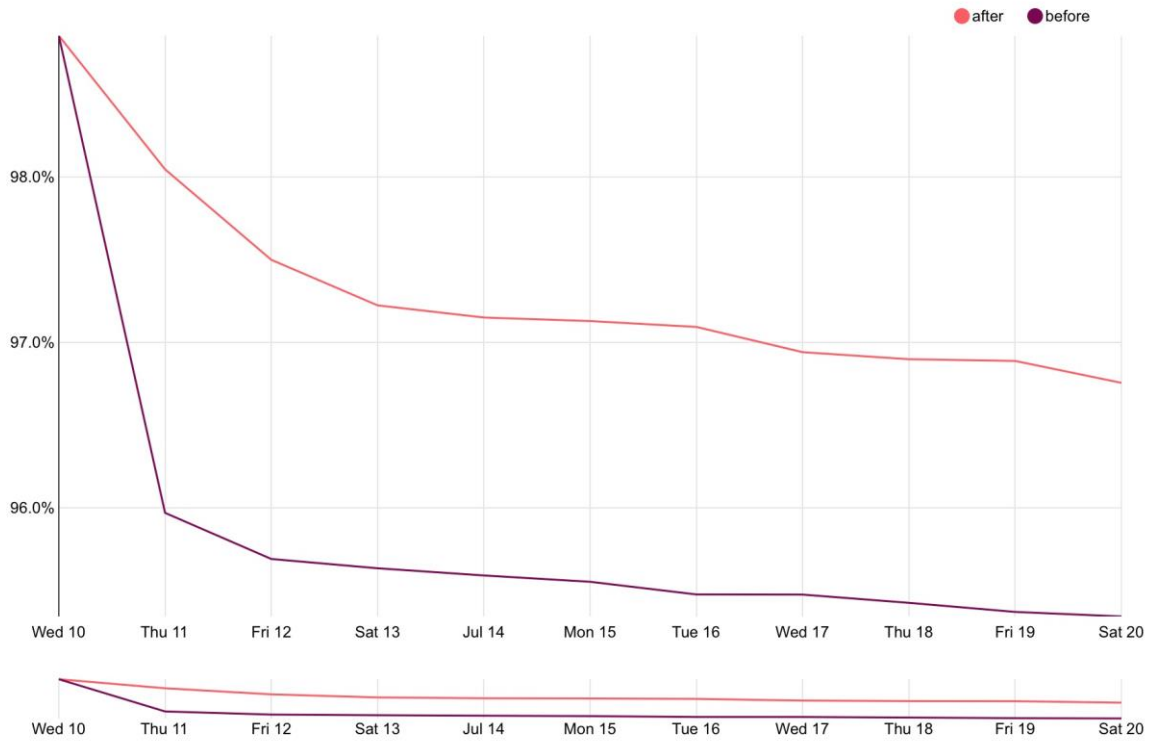

图 svm模型可信度

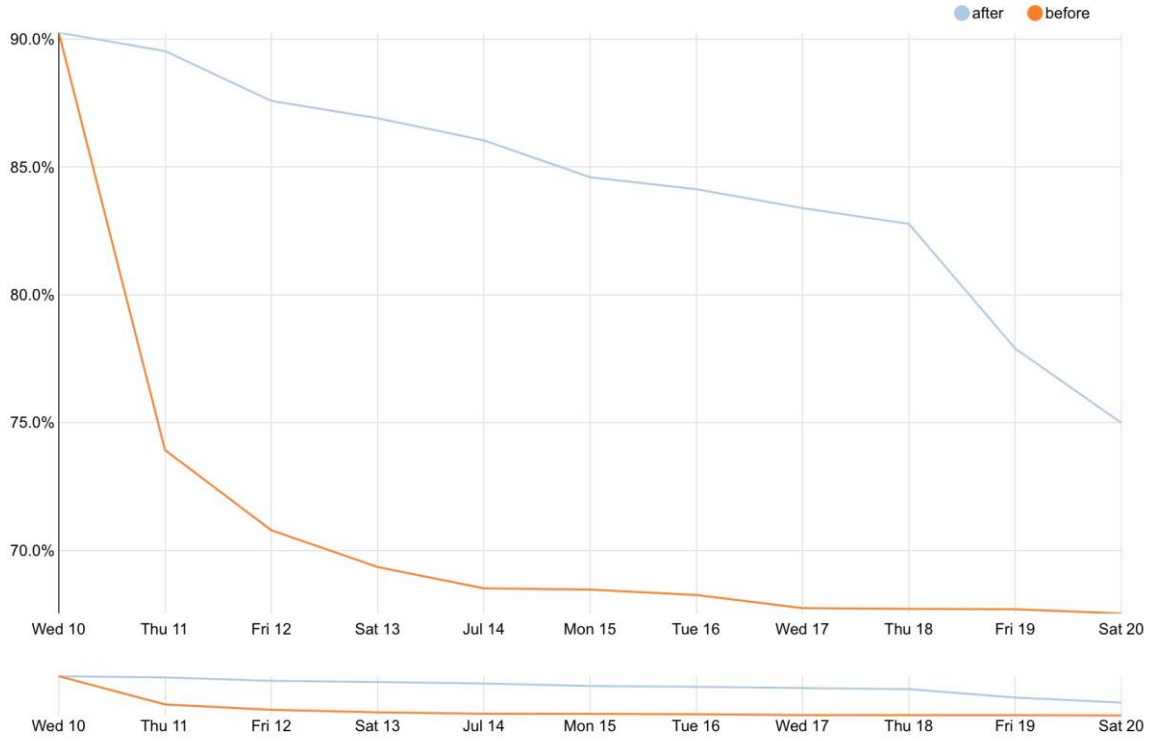

图 XGBoost模型可信度

上面几幅图列举了几个表现较好的单个机器学习模型。可以看出，这些模型的可

信度随时间都会有明显的下降，随机森林和XGBoost模型甚至很快下降到了90%以下；并且，这些模型都是在很短的时间内下降到了较低的水平，随后缓慢下降至稳定。而这些单个模型在经过了情报繁殖之后，可信度较没有繁殖的情况有了显著的提高，不但下降的速度大幅下降，而且下降的时间也有所延后。由此可见，通过情报繁殖，威胁情报库的时效性得以提高，训练出的模型针对新的威胁有更好的识别能力。

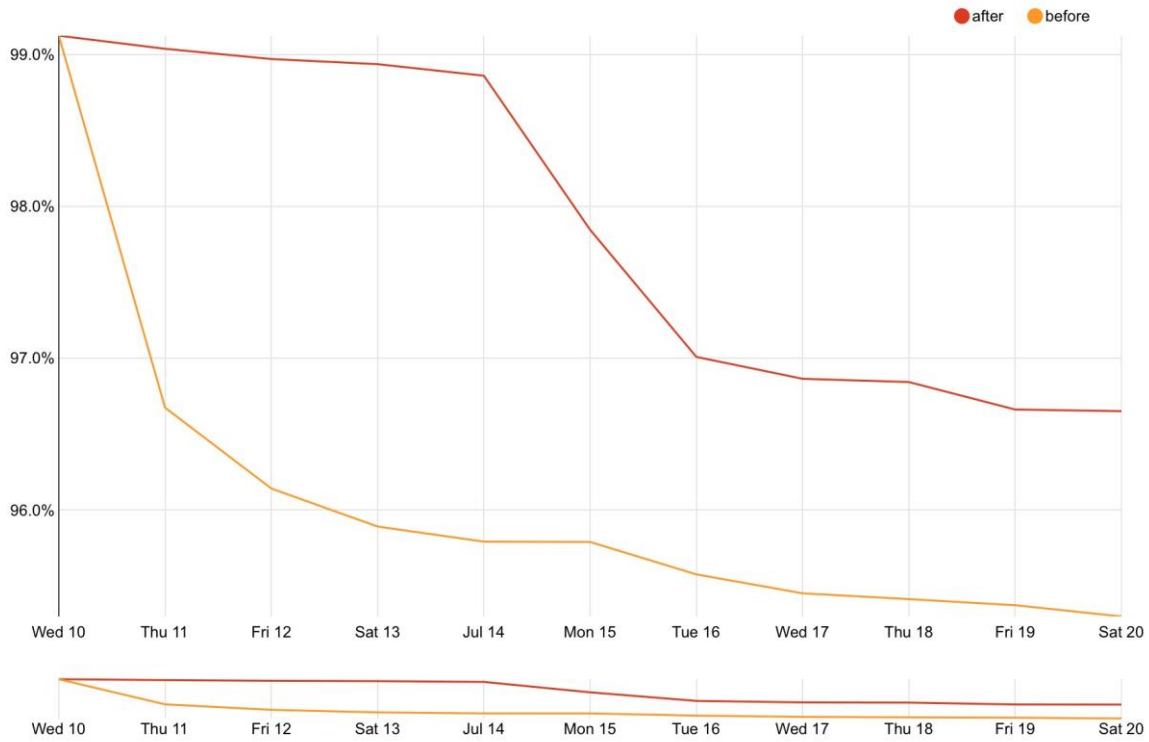

图 融合模型召回率

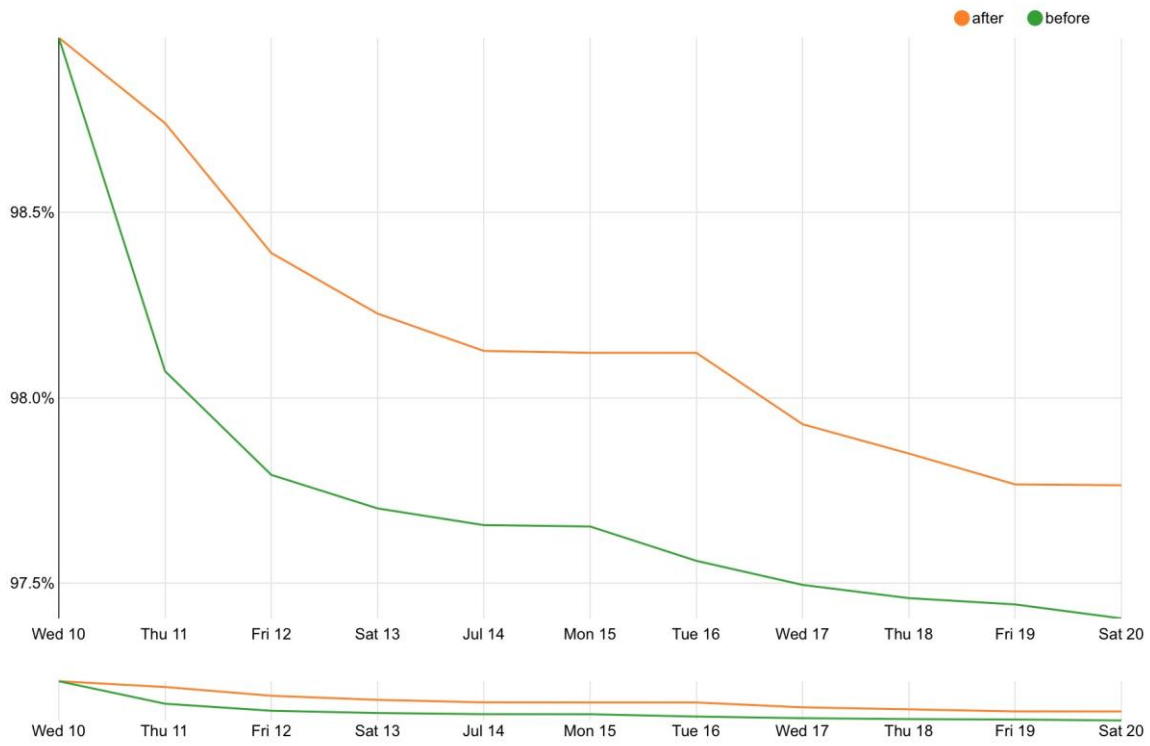

图 融合模型F值

由以上两图可以看出，在进行情报繁殖之前，融合模型的召回率和F值也经过了一个先快速下降到接近最低值，再缓慢下降至稳定的过程，这样的下降趋势非常不利于对威胁情报的识别，因为这会导致模型很快失去应有的精确度。在经过情报繁殖后，融合模型的召回率曲线经过了一个缓慢下降-快速下降-缓慢下降的过程，第一段平稳期为情报库的更新提供了宝贵的时间。而融合模型的F值下降曲线的趋势虽然和繁殖前大致相同，但下降的幅度大大减小，可以领先融合前1个百分点以上。

由上面的数据和分析，我们可以发现，情报繁殖对于我们基于可信度的多模型融合系统的召回率和F值都有明显的提高，在很大程度上缓解了模型的可信度随时间快速下降的情况，为不但扩充了情报库，也为情报库的更新提供了宝贵时间。

3.8 可视化功能测试

Superset 是 Airbnb 开源的数据分析与可视化平台，同时也是由 Python 语言构建的轻量级 BI 系统。Superset 可实现对 TB 量级数据进行处理，兼容常见的数十种关系或非关系型数据库，并在内部实现 SQL 编辑查询等操作。除此之外，基于 Web 服务的 Superset 可实现多用户协使用，并可针对不同角色进行权限管理。

3.7.1 测试方案

将机器学习获得的结果、用户访问网站的相关信息，以及可信度结果以可视化的方式展现出来，首先将对用户访问的网址的打分以统计图的方式展示出来，之后，将用户访问的各个网站的详细信息以表和图的方式表现出来，同时，纵向地将用户在一段时间内访问的网址的信息展示出来。

3.7.2 测试环境搭建

后端: 整个项目的后端是基于 Python 的，用到了 Flask、Pandas、SqlAlchemy。其中:Pandas 用于分析，SQLAlchemy 作为数据库的 ORM、Flask AppBuilder 用作鉴权、规则及 CRUD。

另外，采用 memcache 和 redis 作为缓存，级联超时配置。前端：用到了 npm、react、webpack。Superset 的整个后端是基于 python 开发的，所以我们需要配置 python 的环境，在 python 环境下安装 superset。Superset 目前主要使用 python2.7 跟 python3.4+来进行测 试，推荐使用 python3，不支持 python2.6。本次实验使用 python3.7与Mac OS X Mojave 10.14.4版本的终端进行。

同时利用MySQL搭建本地数据库存储数据，并且与superset进行连接，这样子可以将数据库中存储的表利用superset进行可视化展示。

主要步骤如下:

(1)安装virtualenv用于之后创建虚拟环境：pip install virtualenv

(2)创建一个文件夹，用来存放所有的虚拟环境并且修改相应的环境变量。然后创建python3 的虚拟环境。

(3)在 virtualenv 环境下，使用最新的pip setuptools 库：

pip3 install --upgrade setuptools pip

加密数据库连接信息然后存储到superset的元数据库(SqLite)：

brew install pkg-config libffi openssl python

env LDFLAGS="-L$(brew --prefix openssl)/lib" CFLAGS="-I$(brew --prefix openssl)/include" pip install cryptography==1.9

使用 pip 来安装 superset ： pip install superset

(4)用户注册：fabmanager create-admin --app superset

(5)数据库初始化：superset db upgrade

(6)加载测试数据：superset load_examples

(7)创建默认的角色权限：superset init

(8)本地启动(默认端口 8088)：superset runserver -d

(9)安装依赖文件mysqlclient：

pip install mysqlclient

(10)进入superset的web：http://0.0.0.0:8088/ 配置mysql数据库为数据源

在mac OS X的终端中启动superset如图所示（用virtualenv创建的python3.7环境下的虚拟环境名为py3）：

```
[zhanjingdeMacBook-Pro:~ xueniiii$ workon py3
[(py3) zhanjingdeMacBook-Pro:~ xueniiii$ superset runserver -d
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Starting Superset server in DEBUG mode
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

2019-05-19 21:52:02,023:INFO:werkzeug: * Running on http://0.0.0.0:8088/ (Press
CTRL+C to quit)
2019-05-19 21:52:02,024:INFO:werkzeug: * Restarting with stat
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Starting Superset server in DEBUG mode
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

2019-05-19 21:52:04,602:WARNING:werkzeug: * Debugger is active!
2019-05-19 21:52:04,610:INFO:werkzeug: * Debugger PIN: 256-804-128
```

图 3-27 superset启动

部分可视化结果如下图所示：

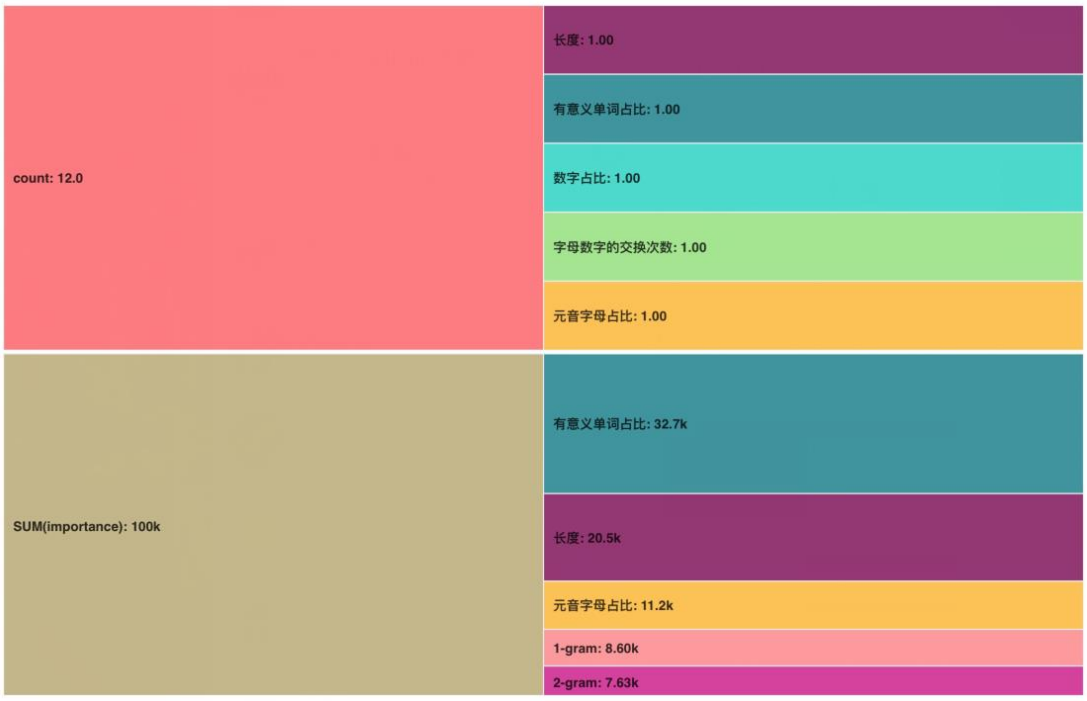

图 3-28 特征值重要性树图

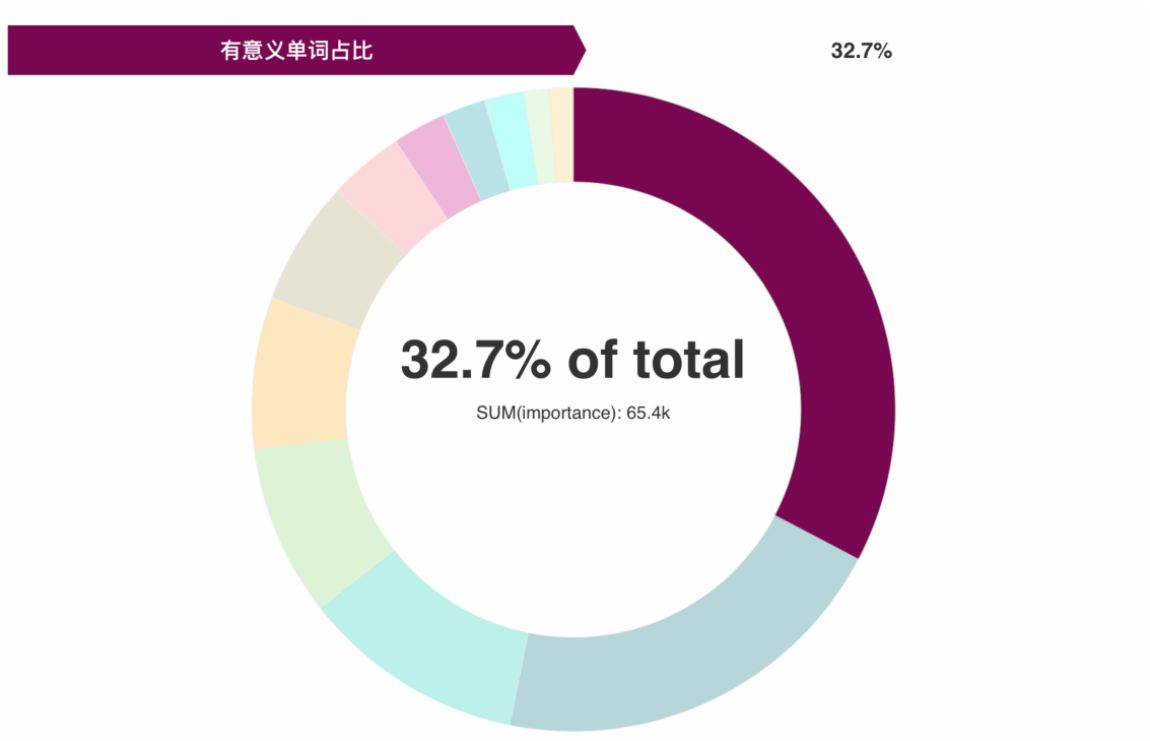

图 3-25 特征值重要性环状图

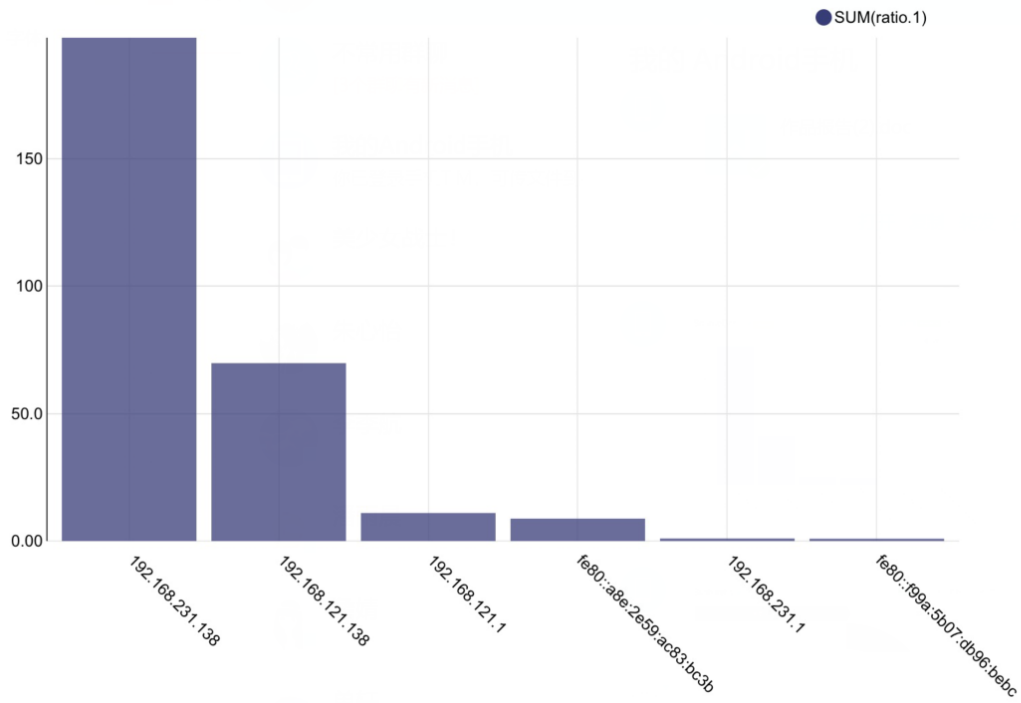

图 3-29 访问源IP柱状图

3.8 本章小结

在本章，我们对作品进行了测试与分析，采用了规范的方法对系统各部分功能和性能进行了测试，确保系统的完整和正常运行。同时也明确了各测试部分间的联动性、实时性和稳定性，对所测试算法进行优化与比对，对测试数据进行了严格的存储和筛选，保证了测试结果的客观性和准确性。

第四章 创新性说明

4.1 基于 zeek/bro 的用户网络行为实时监测

本系统利用实时的网络行为分析器 zeek/bro 对用户访问网站的行为进行监测和分析。当用户对某个网站进行访问，可以获取到该网站的 IP、所属地区、 域名等信息，生成一个具有一定格式的 log 文件进行存储，将此文件的信息作为机器学习的输入，自后续过程中对数据进行进一步的分析处理。

4.2 多模型对大数据进行分析处理

本系统采用多种机器学习算法（XGBoost 算法、 LSTM 算法和朴素贝叶斯算法等），对数据收集系统获取的几百万的威胁情报数据进行机器学习，并利用 IVAPD 对我们建立的机器学习模型进行评估校准。多种机器学习算法同时进行使用，并选取其中可信度较高的结果作为进行融合得到最终结果，这样能够在很大程度上避免了由于某一种机器学习算法对某些特定的威胁情报学习效果不好而导致对于威胁情报的误判，从而大大提高了本系统最终结果的准确度，同时，k 和 rate 两种参数的设置，也能够满足不同用户对于准确率和召回率的不同需求。

4.3 基于可信度的打分方法

本系统一改以往机器学习采用阈值打分的方法，而是基于统计学习的理论，采用一种基于可信度的打分方法。

统计学习由符号学习发展而来，根据符号学习的学习策略可将其分为记忆学习、演绎学习和归纳学习三种方式，其中归纳学习则是指以归纳推理为基础的学习方式，这种学习方式试图从具体实例当中寻找一般规律，统计学习认为这些具体实例满足一定统计学规律，例如独立同分布，但统计学习的诸多理论都来源于统计学研究，例如 VC 维理论、核方法等等。

可信度，也叫做置信度（Confidence），是用来评估机器学习结果的一个指标，是指出现某件事情的时候，另外一些事件也必定出现的概率（针对某种规则而言）。因为分离器对训练集进行分类而得出的准确率和误差率并不能很好反映分类器在未来的工作性能。所以我们需要可信度来衡量我们的分类器的性能。

置信度揭示了 A 出现时，B 是否也会出现或有多大概率出现。如果置信度度为 100%，则 A 和 B 可以捆绑。如果置信度太低，则说明 A 的出现与 B 是否出现关系不

大。

$$confidence(A => B) = P(B|A) = \frac{support(A \bigcup B)}{support(A)} = \frac{support\_count(A \bigcup B)}{support\_count(A)}$$

图 4-1 可信度的公式

4.4 威胁情报的繁殖

本系统对于域名信息数据分析和机器学习模型的预测后，能够得知该域名是否具有威胁性。之后会将被认定为是危险的网站加入之前的黑名单库，更新威胁情报库，实现威胁情报的繁殖。然后再对新的黑名单进行机器学习，重复上述步骤，使得威胁情报库可以不断的进行更新。经由此种方法，我们通过不断的机器学习，不断地获取新的威胁情报，借助原有的威胁情报库得到了一个更大的威胁情报库，而不再是像传统的方法那样，只是对已有的威胁情报库比对，被动地等待情报源提供的滞后的情报。这样做可以提高对威胁情报的利用率，使威胁情报的价值得到更大的发挥。而且当下恶意网站的域名不断更新，生存周期较短，使得很多威胁情报的使用周期也随之变短，很多威胁情报使用过后可能就再无用处。本系对威胁情报繁殖，可以使得整个威胁情报库的情报得到实时地更新，威胁情报库的可用数据增多，也更贴近于当下的威胁情报特征，可用性大大提高，生命周期也在一定程度上得到延长，提高了威胁情报的利用价值。

5.5 实现大数据可视化展示

本系统在对获得的新情报进行分析以及进行威胁情报库的更新后，使用 Superset 平台对数据进行可视化展示，并且将分析的数据进行图形化处理提供给用户。

Superset 是由 Airbnb 开源 BI 数据分析与可视化平台，该工具主要特点是可自助分析、自定义仪表盘、分析结果可视化(导出)、用户/角色权限控制，还集成了一个 SQL 编辑器，可以进行 SQL 编辑查询等，原来是用于支持 Druid 的可视化分析，后面发展为支持很多种关系数据库及大数据计算框架，如：MySQL, Oracle, Postgres, Presto, SQLite, RedShift, Impala, SparkSQL, Greenplum, MSSQL 等。

我们可以通过连接数据库，去对数据库中的单个表进行配置，展示出柱状图，折线图，饼图，气泡图，词汇云，数字，环状层次图，有向图，蛇形图，地图，平行坐

标，热力图，箱线图，树状图，热力图，水平图等图表。

这样我们就可以将分析后的网站信息以图形化的方式直观地展示给用户，便于用户便捷地了解该网站的情况，并且对网站信息和安全状态有一个较为直观的了解和体会。

第五章 总结

本作品提出的基于可信度的网络威胁情报繁殖系统，将传统的基于阈值的判断转化为更加详细的“可信度”和多模型相结合的方式，并利用superset将对应的信息实时显示出来。本作品的技术路线当中，收集全球多个安全公司、安全组织发布的恶意域名，作为系统最初的威胁情报库。然后运用了lstm、xgboost等机器学习算法对大量的威胁情报进行分析处理，之后又利用IVAPD对情报构建的机器学习模型进行进一步分析，得到模型的评分，并且基于可信度对情报进行繁殖。不仅仅提高了威胁情报的利用率，还能够根据可信度来达到繁殖的功能，有助于实现情报复用和构建协同防御体系。最后的可视化展示部分，通过一种直观、实时的方式，让用户可以更加详细的查看威胁情报的具体情况和可信度的分析。

我们今后将从以下几个方面继续改进我们的系统：

1. 扩大威胁情报数据库来源，不仅从安全公司、安全组织的网站上通过爬虫获取威胁情报，还与其他的域名安全性预测系统实现情报共享，建立完善的协同防御体制。
2. 现使用的机器学习模型对于我们的数据拟合速度较慢，我们可以对数据进行特征选择等预处理，提高程序运行速度
3. 尝试更多的机器学习算法，找到可信度更高，更加适合本系统的算法。

参考文献

[1] Ilia Nouretdinov. Inductive Venn-Abers Predictive Distribution. Conformal and Probabilistic Predict ion and Applications,2018

[2] WenboGuo. LEMNA : Explaining Deep Learning based Security Applications. CCS,2018

[3] Vladimir Vovk and Ivan Petej. Venn–Abers Predictors. Proceeding

UAI'14 Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, 2014

[4] Ernst Ahlberg. Using Venn-Abers Predictors to assess Cardio-Vascular Risk. Conformal and Probabi listic Prediction and Applications.2018

[5] Valery Manokhin. Multi-class probabilistic classification using inductive and cross Venn–Abers pred ictors. Conformal and Probabilistic Prediction and Applications. 2017

[6] Chen T, Guestrin C. XGBoost: a scalable tree boosting system. 2016.

[7] T. Chen, S. Singh, B. Taskar, and C. Guestrin. Efficient second-order gradient boosting for condition al random fields. In Proceeding of 18th Artificial Intelligence and Statistics Conference (AISTATS'15), volume 1, 2015.

[8] Ron Bekkerman , Mikhail Bilenko , John Langford, Scaling up Machine Learning: Parallel and Distr ibuted Approaches, Cambridge University Press, New York, NY, 2011

[9] Sun, X.; Luh, P.B.; Cheung, K.W.; Guan, W.; Michel, L.D.; Venkata, S.; Miller, M.T. An efficient a pproach to short-term load forecasting at the distribution level. IEEE Trans. Power Syst. 2016, 31, 2526 –2537

[10] Woodbridge J, Anderson H S, Ahuja A, et al. Predicting Domain Generation Algorithms with Long Short-Term Memory Networks[J]. 2016.

[11] Sak, H.; Senior, A.; Beaufays, F. Long short-term memory recurrent neural network architectures fo r large scale acoustic modeling. In Proceedings of the Fifteenth Annual Conference of the International Speech Communication Association, Singapore, 14–18 September 2014.

[12] Alex Graves. Supervised Sequence Labelling with Recurrent Neural Networks. Studies in Computa tional Intelligence. January 2012

[13] Tianqi Chen. XGBoost: A Scalable Tree Boosting System. the 22nd ACM SIGKDD International C onferenc. August 2016

[14] Xiangrui Meng , Joseph Bradley , Burak Yavuz , Evan Sparks , Shivaram Venkataraman ,

Davies Liu , Jeremy Freeman , DB Tsai , Manish Amde , Sean Owen , Doris Xin , Reynold Xin , Michael J. Franklin , Reza Zadeh , Matei Zaharia , Ameet Talwalkar, MLlib: machine learning in apache spark, The Journal of Machine Learning Research, v.17 n.1, p.1235-1241, January 2016

[15] Niu, D.; Dai, S. A short-term load forecasting model with a modified particle swarm optimization algorithm and least squares support vector machine based on the denoising method of empirical mode decomposition and grey relational analysis. Energies, 10, 408, 2017