北京交通大学

硕士专业学位论文

DGA 恶意域名检测与评估系统的研究与实现

Research and Implementation of a DGA Malicious Domain Detection and Evaluation System

作者：李美慧

导师：常晓林

北京交通大学

2024 年 6 月

学校代码：10004 密级：公开

北京交通大学

硕士专业学位论文

DGA 恶意域名检测与评估系统的研究与实现

Research and Implementation of a DGA Malicious Domain Detection and Evaluation System

作者姓名：李美慧 学 号：22140488

导师姓名：常晓林 职 称：教授

专业学位类别（领域）：软件工程 学位级别：硕士

北京交通大学

2024 年 6 月

摘要

在互联网技术迅猛发展的背景下，网络空间既为人类活动提供了前所未有的便捷性，同时也伴随着日益严峻的网络安全挑战。特别是，僵尸网络作为一种具有显著影响力和破坏力的网络攻击手段，其采用 DGA（Domain Generated Algorithm）技术，使得攻击者能够通过被控制的主机访问到由C&C（命令和控制）服务器注册的恶意域名，进而与这些主机建立通信，并实施命令控制或数据窃取等恶意行为。由于DGA生成的域名具有高度随机性，攻击者得以在频繁变化的域名请求中隐藏自身，使得传统的基于黑名单的安全机制难以有效拦截。在政府、医疗等关键机构的应用场景中，对DGA技术的检测尤为关键，这些机构通常对数据隐私有极高的要求，且由于内部数据不能上传至云端进行分析，因此迫切需要能够在数据不出网的前提下，实现实时检测的系统，以便对内部网络进行有效监控。此外，这些场景对检测系统的稳定性和效率提出了更高的要求，需要一种轻量级的系统，能够在海量的网络数据中有效识别DGA恶意域名。针对上述特定场景，设计并实现了一套 DGA 恶意域名检测与评估系统，本文主要工作如下：

（1）提出了一种基于机器学习的 DGA 域名检测方法。该方法通过大量实验数据，对恶意域名与合法域名的特征进行相关性分析，提取并验证这些特征的有效性。并根据相同家族的DGA域名相似性和域名本身的随机性，引入了两阶段检测流程。实验结果表明，本文提供的检测方法在误报率和检测效率上有较好的表现，适用于上述目标场景。

（2）提出了一种 DGA 检测评估框架。现有的评估指标大多是对算法的效果说明，缺少反馈机制，且本系统采用线下部署形式难以持续更新。为解决上述问题，本系统设计了评估框架并引入运营评估中心。该中心依据系统在实际应用场景中运营人员所关注的特定指标，构建了一套评估体系，实时更新评估数据，反馈给检测模型做持续优化。该框架通过多组数据集验证，可以帮助运营人员定期优化检测效果并动态展示评估结果，确保其在实际应用中的有效性和可靠性。

（3）设计并开发了基于大数据平台的 DGA 检测与评估系统。该系统基于分布式大数据平台实现。在检测部分，采用了 Apache Spark 这一高性能的分布式数据处理框架，有效实现了对大规模数据的并行处理。在存储部分，引入了Hadoop的分布式存储系统，实现基于云下的本地部署。在设计思路上，采用可扩展的组件开发技术为后续进一步提升系统的检测范围提供扩展能力。在完成系统的设计与开发后，采用科学测试方法，验证了本系统可以联动边界设备，完成流量及日志的检测评估以及前台展示，有效协助运营人员维护网络环境。

关键词：恶意域名；机器学习；随机森林；大数据

ABSTRACT

Against the backdrop of the rapid development of Internet technology, cyberspace has provided unparalleled convenience for human activities while also posing increasingly severe cybersecurity challenges. In particular, botnets, as a form of cyberattack with significant influence and destructive power, employ DGA (Domain Generated Algorithm) technology, enabling attackers to access malicious domains registered on C&C (Command and Control) servers through compromised hosts. This allows them to establish communication with these hosts and carry out malicious activities such as command and control or data theft. Due to the high randomness of DGA-generated domains, attackers can conceal themselves amidst frequent changes in domain requests, making traditional blacklist-based security mechanisms ineffective in intercepting them. In the application scenarios of key institutions such as government and healthcare, the detection of DGA technology is particularly critical. These institutions typically have stringent requirements for data privacy and, since internal data cannot be uploaded to the cloud for analysis, there is an urgent need for a system that can perform real-time detection without data leaving the network, effectively monitoring internal network traffic. Furthermore, these scenarios demand higher stability and efficiency from the detection system, necessitating a lightweight system capable of effectively identifying DGA malicious domains amidst massive network data. In response to these specific scenarios, a DGA malicious domain detection and evaluation system has been designed and implemented, with the main contributions of this work as follows:

(1) A DGA domain detection method based on machine learning is proposed. This method analyzes the correlation between the features of malicious and legitimate domains using a large amount of experimental data and extracts and verifies the effectiveness of these features. Considering the similarity of DGA domains within the same family and the randomness of domain names themselves, a two-stage detection process is introduced. Experimental results show that the detection method provided in this paper has good performance in terms of false positive rate and detection efficiency, and is suitable for the above target scenarios.

(2) A DGA detection evaluation framework has been proposed. Existing evaluation metrics primarily describe the effects of algorithms, lacking a feedback mechanism. Furthermore, the offline deployment of this system makes it challenging to continuously update. To address these issues, the system has been designed with an evaluation framework

and an operational evaluation center. The center constructs an evaluation system based on specific indicators that are of concern to the operational personnel in real-world application scenarios. It updates the evaluation data in real-time and provides feedback to the detection model for continuous optimization. This framework has been validated with multiple datasets, helping operational personnel to regularly optimize detection effectiveness and dynamically display evaluation results, ensuring its effectiveness and reliability in practical applications.

(3) A DGA detection and evaluation system based on a big data platform is designed and developed. The system is implemented based on a distributed big data platform. In the detection part, the high-performance distributed data processing framework Apache Spark is used, effectively achieving parallel processing of large-scale data. In the storage part, the distributed storage system of Hadoop is introduced to achieve local deployment based on cloud computing. In terms of design, scalable component development technology is used to provide expansion capabilities for further enhancing the detection scope of the system in the future. After completing the design and development of the system, scientific testing methods are used to verify that the system can link with boundary devices, complete the detection and evaluation of traffic and logs, and display on the front end, effectively assisting in-network operation personnel in maintaining the network environment.

KEYWORDS: Malicious Domains; Machine Learning; Random Forest; Big Data

目　录

1 绪论

1.1 研究背景及意义

在当今社会中，网络已经成为了人们生活、工作中不可或缺的一部分。无论是个人还是企业，都离不开网络的支撑。然而，随着网络的普及，硬件和软件也在不断的迭代更新，网络安全相关问题逐渐凸显，给用户的信息与财产安全带来了严重威胁。在众多网络安全问题中，DGA（Domain Generation Algorithm）恶意域名检测是一个重要的研究方向。并且在机器学习领域取得显著成就的背景下，DGA 域名检测技术正经历着快速的优化和革新，这些进步不仅加快了技术发展的步伐，还为网络安全领域带来了创新的解决方案。

在互联网的迅速发展的当下。它为人们提供了丰富的信息资源，使得人们可以随时随地获取所需的信息。网络也为人们的生活和工作带来了极大的便利。在社交娱乐、在线购物、远程办公、在线教育等方面，网络已经深入到个人生活的方方面面。此外，网络也是企业运营的重要支撑，企业通过网络进行营销、客户服务、供应链管理等，提高了运营效率和竞争力。

随着网络的普及，网络安全问题日益严峻。主要的安全问题包括计算机病毒、恶意软件、网络钓鱼和黑客攻击等[1]。这些问题对用户的信息安全和财产安全构成严重威胁。例如，计算机病毒可通过感染文件、软件等方式传播，导致系统崩溃、数据丢失等问题；而恶意软件可能窃取用户的个人信息、银行账号等敏感信息，给用户带来经济损失；钓鱼网站则是一种利用伪造的网站或邮件欺骗用户输入个人信息的行为，同样会带来严重后果。网络安全对个人及企业都非常重要。首先，网络安全关系到用户的信息安全。在数字化时代，个人信息具有重要价值，一旦泄露，可能导致个人隐私被侵犯、财产被盗取等问题。其次，网络安全关系到用户的财产安全。网络犯罪分子通过恶意软件、网络钓鱼等手段，可以轻易地盗取用户的银行账号、密码等敏感信息，给用户造成经济损失。此外，网络安全还关系到用户的正常使用。网络攻击、系统漏洞等问题可能导致用户无法正常访问网络资源，影响用户的工作和生活。

在网络安全领域，恶意软件的传播和控制是一个持续性的威胁，而域名生成算法（DGA）作为恶意软件逃避检测的一种手段，已经成为网络安全研究的重要课题。恶意软件通过 DGA 生成大量随机的域名[2]，这些域名被用于与命令和控制（C&C）服务器进行通信，从而实现恶意活动的指挥和控制。攻击者通常会将这

些域名注册在多个不同的顶级域下，以增加追踪和封禁的难度。历史上，已经发生了多起利用 DGA 技术的恶意软件攻击事件。例如，Conficker 病毒就曾利用 DGA 生成了数以万计的域名，用于与其 C&C 服务器通信，感染了全球数百万台计算机。同样，ZeuS 银行木马也使用 DGA 来定期更换与其 C&C 服务器通信的域名，以逃避安全软件的检测。这些攻击事件不仅造成了巨大的经济损失，还对全球网络安全构成了严重威胁。

传统的 DGA 检测方法主要依赖于黑名单和白名单机制，这些方法在面对快速变化的 DGA 时无法达到预期效果。黑名单方法需要不断地更新和维护，才可以包含最新的恶意域名，但这种方法容易受到时间延迟的影响，导致无法及时识别新生成的恶意域名。白名单方法则相对保守，可能会误判一些合法的动态域名。此外，传统的基于特征的检测方法也面临着检测准确性和效率之间的平衡问题。

随着命令和控制（C&C）服务器利用 DGA 技术生成恶意域名的策略不断进化和演变，网络攻击事件日益增多，对社会稳定和安全的威胁日益严重。研制一种能够检测和评估 DGA 恶意域名的系统显得尤为重要。通过结合机器学习和数据分析技术，可以实现对 DGA 域名的识别和预测，从而提高检测的准确性和效率。有效地识别和迅速应对这些 DGA 生成的恶意域名是应对这些网络威胁的关键所在。近年来，随着机器学习相关技术的成熟，在追求对 DGA 恶意域名的高检查率外，对检测算法的性能和使用场景也有了更多的要求。因此，在当前网络环境下，探索具有较短检测周期、较低资源消耗的 DGA 恶意域名检测方法显得尤为迫切和必要。通过研制 DGA 恶意域名检测与评估系统，使其在特定场景下，可以为网络安全防护提供一种更为有效和先进的工具。这不仅能够帮助企业和个人用户减少因恶意软件攻击而造成的损失，还能够提升整个网络环境的安全性和可靠性。此外，通过对 DGA 技术的深入研究，可以为网络安全防护提供更为全面的策略和方法，推动网络安全技术的发展和创新。

1.2 国内外研究现状

如前所述，在网络安全领域，域名生成算法（DGA）的检测研究经历了从传统算法到机器学习，再到传统机器学习和深度学习模型结合的发展过程。DGA 是一种被恶意软件广泛采用的域名生成技术，通过算法生成大量随机的、难以预测的域名，用于与命令和控制（C&C）服务器进行通信。这些恶意域名往往与僵尸网络、恶意软件传播、网络攻击等恶意行为相关联。因此，针对 DGA 恶意域名生成算法和检测算法的研究对于保护网络安全具有重要意义。

1.2.1 DGA 算法

DGA 的概念最早在 2007 年由 Paul Royal 和 J. Alex Halderman 等人提出，他们在针对 Storm Worm 恶意软件的研究中首次描述了这种算法。DGA 的基本原理是利用种子（如当前日期、域名散列值等）生成看似随机但实际上可预测的域名。通常是由恶意软件在感染目标主机后，DGA 恶意域名在其本地生成一组域名，然后尝试与这些域名进行通信，使被攻陷主机进一步接收并执行控制和指令。这种攻击方式难以防御的原因在于，DGA 生成的域名具有高度随机性和动态性，使得传统的基于黑名单和白名单的防御机制难以有效应对。在之后的发展中，DGA 恶意软件也通常会采用加密通信、域名轮换等手段，进一步增加了检测和防御的难度。

在 2016 年， Plohmann[1]等人深入探讨了 DGA（域名生成算法）域名的生成机制及其检测技术，并强调了这一领域对于对抗僵尸网络攻击的至关重要的作用。Plohmann 等人对僵尸网络中，已经发现的 DGA 进行了深入研究。通过对 43 个僵尸网络环境产生的流量进行分析，发现其中 23 个僵尸网络完全依赖 DGA 作为其命令与控制（C&C）服务器集合的核心手段。这些僵尸网络利用域生成算法自动产生大量看似随机的域名，以此来隐蔽它们的 C&C 通信。攻击者与被感染的肉鸡通过相同的生成算法，生成了同一套域名列表。通过分析这些 DGA 家族的算法代码，研究人员提前计算出了潜在的域名，总计超过了 1800 万个。他们宣称，通过这种方法可以预测并识别未来的 DGA 域名，从而有效追踪恶意软件及其相关行为。随着网络安全专家的不懈努力，越来越多的 DGA 算法机理被揭示出来。在此背景下，360 网络实验室已经公布了超过 60 个 DGA 家族及其变种的算法核心逻辑。

在网络安全领域还推出了一些创新的 DGA 算法。例如，2016 年，Anderson H 等人提出了一种名为 DeepDGA 的算法[3]，该算法利用生成 GAN 对抗网络，通过多轮迭代生成对深度学习网络更具挑战性的 DGA 域名；同时，这些域名也被用作训练样本，以提高检测模型对未知 DGA 域名的识别能力。2019 年，Peck J 等人发布了一种新型的 DGA 算法 CharBot，它生成的域名能够逃过当时先进的检测技术，包括基于手工特征的随机森林和长短期记忆网络（LSTM）方法，且这种算法在没有目标 DGA 分类器知识的情况下仍具有出色的逃逸能力。同年，Yun Xiaochun 等人[4]结合神经语言模型和生成对抗网络，提出了一种名为 Khao 的新算法，通过混合首字母缩略词和音节来模拟真实域名。到了 2023 年，Zhai Yo 等人提出了一种名为 BadDGA 的算法，它专门针对基于 LSTM 的域名检测方法，提供了一种后门攻击方法，并展示了其攻击的高成功率。

这些研究表明，DGA 算法正在不断进化，DGA 生成的恶意域名可以逃避传统的检测机制，增加了 DGA 恶意域名被检出的概率。

1.2.2 DGA 恶意域名检测方法

首个 DGA 检测研究是由 Hussey 等人于 2007 年提出的，他们通过分析 DGA 生成的域名特征，提出了一种基于字典树的检测方法。该方法构建一个字典树，将已知的正常域名和恶意域名分别存储在字典树的两侧，然后通过比较待检测域名与字典树的匹配程度来判断其是否为恶意域名。2011 年，Kurt Thomas 等人发表了关于 DGA 域名的检测研究，他们提出了一种基于频率的检测方法，通过分析域名的字符分布和 N-gram 频率来识别 DGA 域名。但对未知 DGA 算法的检测效果不佳。

随着研究发展，机器学习逐渐被应用起来，图 1-1 展示了基于机器学习的 DGA 域名检测算法的分类。2011 年，Kolter 和 Mironov 提出了一种基于决策树的 DGA 检测方法，通过对大量域名样本进行特征提取和分类，实现了对 DGA 域名的有效识别。这种方法具有较高的检测准确率，但需要大量的训练数据和时间来构建决策树模型。

Satoh 等人评估了 755 个含有恶意流量的 DNS 跟踪和 756 个真实 DNS 服务器的流量。但是该研究并未提供对其他 DGA 僵尸网络数据集的评估结果[5]。有团队基于大数据平台下的系统架构，整合了多种算法分析，分阶段对 DGA 域名进行检测[6] 。Bilge 等研究者利用域名地址数量等特性，通过对 DNS 流量异常的监控来识别 DGA 域名。然而，由于恶意域名持续进化，缺乏充足的训练样本导致模型训练不够稳定[7]。

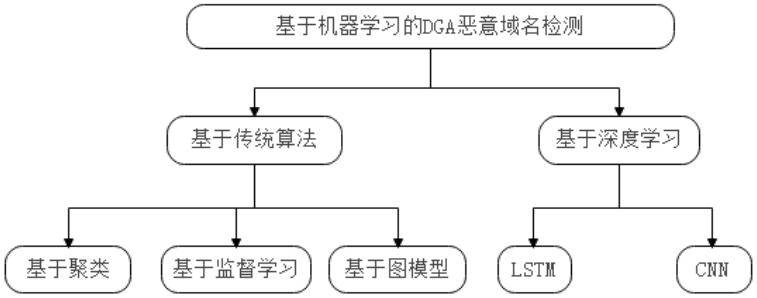

图 1-1 DGA 恶意域名检测算法的分类

Figure 1-1 Classification of DGA Malicious Domain Name Detection Algorithm

2017 年，Mac 等研究人员评估在 DGA 检测中使用有监督的机器学习的效果。

他们开发了包括决策树在内的多种模型，并测试了 SVM、CNN-LSTM 等检测模型的性能。研究使用了 Alexa 良性域名作为非 DGA 标记的数据，以及 Bambenek Consulting 提供的 37 个 DGA 域家族作为 DGA 标记的数据，共 81490 个 DGA 域名。实验结果显示，SVM 和 LSTM 模型的二分类准确率超过 99.55%，优于传统机器学习模型。然而，在多分类任务中，结果并不理想，存在多个未被检测出的 DGA 家族。

近年来，随着深度学习技术的快速发展，基于深度学习的DGA检测方法逐渐成为研究的热点。在运用机器学习技术的背景下，这些方法通常涉及特征的提取以及利用分类器进行恶意域名生成检测。早期研究汇总了 44 个特性以区分恶意和良性域名。普遍认为，机器学习方法利用的特征数量越多，其准确率越高。但是，DNS流量往往非常巨大，针对每条数据，均对所有 44 个特征进行提取会导致检测的时间过长。在大数据的环境下，每分钟可能产生数以百万计的 NXDomain，而对所有域名提取全部的 44 个特征将耗费大量计算资源，这在实际应用中是不切实际的。为了提高特征提取的效率，可以采用自动特征选择技术，有研究如 Truong 等人引入了双字符模型，基于字符串完成 DGA 域名检测[8]。同样，Davuth 等人利用SVM构建分类器，通过筛选词频率来检测恶意域名。这些方法都在努力平衡特征提取的全面性和计算的可行性[9]。2016 年，Plohmann 等人提出了一种基于深度神经网络的DGA检测方法，通过对域名样本进行嵌入向量表示，然后输入到神经网络中进行分类。这种方法在准确率和效率上都取得了较好的表现，但需要大量的训练数据和计算资源。

Woodbridge[10]等研究者运用了长短期记忆网络（LSTM），将输入的字符串转换为基于字符级的词向量，然后通过 LSTM 层提取时序特征。并且将提取的特征利用逻辑回归分类算法进行分类。在DGA域名的二分类任务中，该方法展现出了卓越的识别准确率。在多分类任务中，该方法同样显示出优越性。然而，由于 DGA 域名训练数据源自网络环境，不同僵尸网络家族的 DGA 域名数量占比存在较大差异，导致家族类别失衡。Anderson[11]和 Yin[12]等人采用生成对抗网络（GAN）来缓解这一问题，将不同家族类别的数据作为对抗网络的输入，拟合数据分布，将生成新的域名数据也补充到数据集中，有效提升了多分类下的检测性能。在 2022 年的研究中，Tong Anh Tuan 等人基于 LSTM 与注意力机制，针对 DGA 域名的二分类和多分类问题，分别提出了 LA_Bin07 和 LA_Mul07 两个模型，提升了检测精度。他们还发现，不同DGA家族的上下文不同，可以根据域名关键字的特征来辅助分类。Sarojini S 等人提出了一种多头自注意-递归卷积神经网络-自注意双向长短期记忆模型的深度学习方法，提升了对基于单词生成的DGA域名的检测性能。Cheng Yang 等人提出了一种基于 N-gram 算法和 Transformer 模型并

行的检测模型，能够较好地捕获字母组合特征和位置特征，提高了对于随机字符型 DGA 域名的检测效果，但对随机单词和随机音节 DGA 域名的检测性能还有待提升。Huang Weiqing 等人为了提高模型对于小样本域名数据集的检测性能，提出一种名为 PEPC 的深度学习模型，由预训练嵌入模块和深度卷积神经网络模块组成。其中预训练嵌入模块基于 BERT 模型，通过对大规模未标记的开源域名数据集进行自我监督训练来生成域名嵌入。该模型在小训练样本中实现了较好的检测性能，但是其预训练步骤的计算量较大、计算条件要求较高。在针对单词型 DGA 域名检测的研究工作中，Zhai You 等人提出了一种基于表征融合的检测方法，融合了域名的上下文模型提取的特征和基于图算法提取的特征，结果显示表征融合方法对于提升单词型DGA域名的检测性能是有效的。但是上述方法主要针对随机单词型 DGA 域名的分类，而没有包含其他两种 DGA 域名分类的实现。

Yu[13]采用卷积神经网络（CNN）检测 DGA 域名。CNN 在处理文本类数据时，通常采用一维卷积，通过在向量化的字符上滑动卷积窗口，获取域名的不同特征，CNN 在局部特征获取上具有较好的效果。多个研究均对 CNN 和 LSTM 在 DGA 域名检测任务中的表现进行了对比。文献[14]和文献[15]分别评估了 CNN、LSTM 以及仅使用域名字符串作为输入的监督学习方法，并探讨了 CNN 与 LSTM 的结合。实验证明，CNN-LSTM 深度学习架构在性能上最为出色[16]，这得益于 CNN 在发现局部特征方面的能力和 LSTM 在处理长期时序依赖方面的优势。

综上所述，DGA 检测的研究经历了从传统算法到机器学习，再到传统和深度学习模型结合的发展过程。尽管近年来基于深度学习的DGA检测方法取得了显著的进展，但仍存在一些挑战和瓶颈。在深度学习的DGA域名检测研究中，早期方法主要依赖于循环神经网络（RNN）作为基础架构，对那些具有高随机性的DGA域名展现出较好的检测能力。然而，这些方法在处理基于词典生成的DGA域名时效果欠佳。随着研究的深入， Pereira 等人[17]则计了一种图形学习方法，用以研究基于单词列表的 DGA 所使用的单词列表。Yang 等人研发了一种随机森林分类器[18]，这一分类器通过词频、元辅音比例等特性对基于单词列表的 DGA 进行分类。总体而言，检测DGA域名的挑战主要集中在大数据的场景下，如何高性能的有效识别 DGA 域名上。

1.3 主要研究内容

本文为了实现 DGA 恶意域名的检测与评估系统，主要的工作内容如下：

（1） 提出了一种基于机器学习的DGA域名检测方法。该方法通过大量实验数据，对恶意域名与合法域名的特征进行相关性分析，提取并验证这些特征的有效

性。并根据相同家族的DGA域名相似性和域名本身的随机性，引入了两阶段检测流程。实验结果表明，本文提供的检测方法在误报率和检测效率上有较好的表现，适用于上述目标场景。

（2） 提出了一种 DGA 检测评估框架。现有的评估指标大多是对算法的效果说明，缺少反馈机制，且本系统采用线下部署形式难以持续更新。为解决上述问题，本系统设计了评估框架并引入运营评估中心。该中心依据系统在实际应用场景中运营人员所关注的特定指标，构建了一套评估体系，实时更新评估数据，反馈给检测模型做持续优化。该框架通过多组数据集验证，可以帮助运营人员定期优化检测效果并动态展示评估结果，确保其在实际应用中的有效性和可靠性。

（3） 设计并开发了基于大数据平台的 DGA 检测与评估系统。该系统基于分布式大数据平台实现。在检测部分，采用了 Apache Spark 这一高性能的分布式数据处理框架，有效实现了对大规模数据的并行处理。在存储部分，引入了Hadoop的分布式存储系统，实现基于云下的本地部署。在设计思路上，采用可扩展的组件开发技术为后续进一步提升系统的检测范围提供扩展能力。在完成系统的设计与开发后，采用科学测试方法，验证了本系统可以联动边界设备，完成流量及日志的检测评估以及前台展示，有效协助运营人员维护网络环境。

1.4 论文组织结构

本文的研究目的是研制适用于目标场景的 DGA 恶意域名检测评估系统，设计、实现系统并验证其可行性。根据上述目的，本论文的结构可以分为六个章节：

第一章绪论部分阐述了研究的背景和意义，探讨了网络安全领域，特别是 DGA 恶意域名问题的重要性。并对国内外在 DGA 算法及其检测方面的研究现状进行了全面的文献研究，并详细介绍了本论文的核心研究内容和预期贡献。

第二章介绍了相关的理论和技术基础，包括与DGA相关的多个方面内容。第一节详细描述了僵尸网络的攻击机制及其对网络安全的威胁。第二节阐明了 DNS 服务器的基本原理和作用，为理解DGA的工作机制提供了必要的基础。第三节重点介绍了DGA的概念及其生成算法。第四节则介绍了检测算法中使用的机器学习理论，包括聚类算法、决策树和随机森林生成算法等。

第三章聚焦于DGA恶意域名检测方法的设计与实现。第一节概述了检测方法的整体设计，引入了两阶段检测流程。第二节基于前文提出的检测方法，通过大量数据分析了合法域名与DGA域名的区别，根据期望效果，选取流程中使用的算法和特征，并介绍了具体实现。最后一节首先说明了实验环境和数据集，最后针对实验结果进行了对比分析，实验结果表明整体检测方法符合预期效果。

第四章提出了一种 DGA 恶意域名的评估框架。首先在本系统中引入运营中心，并基于目标场景设计评估框架。其次介绍了评估框架的具体实现，评估框架分为特征相关性评估和检测性能评估两部分，并分别实现了评估结果的可视化展示。最后，针对评估框架进行实验验证，结果表明此框架可以有效协助运营人员完成对检测模型的定期评估及检测方法的持续优化。

第五章基于前述章节的理论和实践成果，开展了DGA恶意域名检测系统的实现与验证。这一章节包括了系统的需求分析、架构设计、模块设计与实现，以及系统的测试等关键环节。

第六章对整个研究进行了总结，并对未来的研究方向进行了展望。在总结部分，本文对研究内容进行了全面回顾，并对关键结果进行了分析。在展望部分，本文根据当前系统的实际应用情况，提出了未来可能的改进和完善方向。

2 相关理论及技术

2.1 僵尸网络

僵尸网络，也称为 Botnet，是一种由大量被黑客控制的感染了恶意软件的计算机和其他设备组成的网络。这些设备通常被称为“僵尸”或“僵尸机”，它们被远程操控执行各种恶意活动，而设备的所有者往往对此毫无察觉。僵尸网络是网络犯罪的一种形式，对个人、组织乃至国家安全构成严重威胁。

僵尸网络的构成主要包括三个部分：僵尸主控（Botmaster 或 Botherder）、僵尸程序（Bot）和命令和控制服务器（C&C 服务器）。僵尸主控是僵尸网络的创建者和操控者，他们通过远程控制来指挥僵尸网络执行特定任务[19]。僵尸程序是植入受害者设备中的恶意软件，让它成为僵尸网络的一部分。这些程序通常通过社交工程、软件漏洞或弱密码等方式传播。命令和控制服务器是僵尸网络的中枢，用于接收来自僵尸主控的指令，并向下发指令给僵尸机。

僵尸网络作为一种强大的网络犯罪工具，其恶意活动形式多样，包括分布式拒绝服务攻击（DDoS）、信息窃取、垃圾邮件发送、点击欺诈和传播恶意软件等。在 DDoS 攻击中，僵尸网络通过向目标发送大量请求，使其服务瘫痪。信息窃取方面，它能够盗取个人敏感信息和企业的机密数据。垃圾邮件发送则用于诈骗或传播恶意软件，而点击欺诈则涉及自动点击在线广告以欺骗广告商。此外，僵尸网络还可以用来传播其他类型的恶意软件，如病毒、木马等[20]。

僵尸网络构成了互联网信息系统的一个普遍威胁，它能够实施拒绝服务攻击，并广泛传播含有恶意目的的垃圾信息[21]。在网络安全领域中，检测和防御僵尸网络至关重要。已有研究提出了多种僵尸网络检测方法，包括基于异常检测、基于签名识别以及利用 HoneyNet 技术的方法[22]。Tong、Long、Taniar[23]提出了一种创新方案，用来检测和分类由域生成算法（DGA）导致的僵尸网络实例，他们通过结合长短期记忆网络（LSTM）和注意力机制，提出了两个深度学习模型，这两个模型可以有效二分类 DGA 域名并检测出此域名对应的家族名称。实验证明，这些模型在处理 DGA 僵尸网络的二元和多类分类问题上具有极高的准确性。

在僵尸网络发挥作用的关键是与命令和控制（C&C）服务器的通信，需要知道 C&C 服务器的 IP 地址才能建立连接[24]。直接使用固定 IP 地址的 C&C 服务器容易受到防火墙、入侵检测和预防系统（IDS/IPS）等安全措施的拦截，如黑名单和白名单的过滤机制。为了解决这个问题，僵尸程序采用了域名和多宿主技术，

作为隐蔽的查询 IP 地址和命令和控制（C&C）服务器的解决方法。域名和 IP 地址是一对多的关系，多宿主机技术可以实现多个域名信息对应同一个IP地址[25]。但是，这种方法逻辑简单，容易被破解，所以并没有推广使用。所以，基于域名生成算法的 DGA 恶意域名被引入到实际的攻击活动中。C&C 服务器和肉鸡采用同一种生成算法生成域名列表。在这个列表中，部分域名会指向攻击者指定的 IP 地址。这种方法相比之前的方法有了显著改进，因为它允许被控主机查询大量的域名从而获取 C&C 服务器 IP，有效降低了攻击活动被检测出的概率，同时 C&C 服务器容易被攻击者隐藏起来，而被控主机因为与服务器拥有同一张域名表，当攻击者更改 C&C 服务器 IP，受控的肉鸡仍然可以轻易的找到他们。僵尸网络的检测问题主要是判断域名的性质，判定域名是合法域名还是DGA域名，同时这个问题也可以扩展为一个多分类问题，以准确识别恶意域名所属的域名家族。迅速的发现网络中存在的DGA攻击，即使它们已经感染了计算机，但是尽早遏制僵尸网络的发展，可以有效降低损失和风险。DGA 的攻击原理及网络模型如图 2-1 所示。

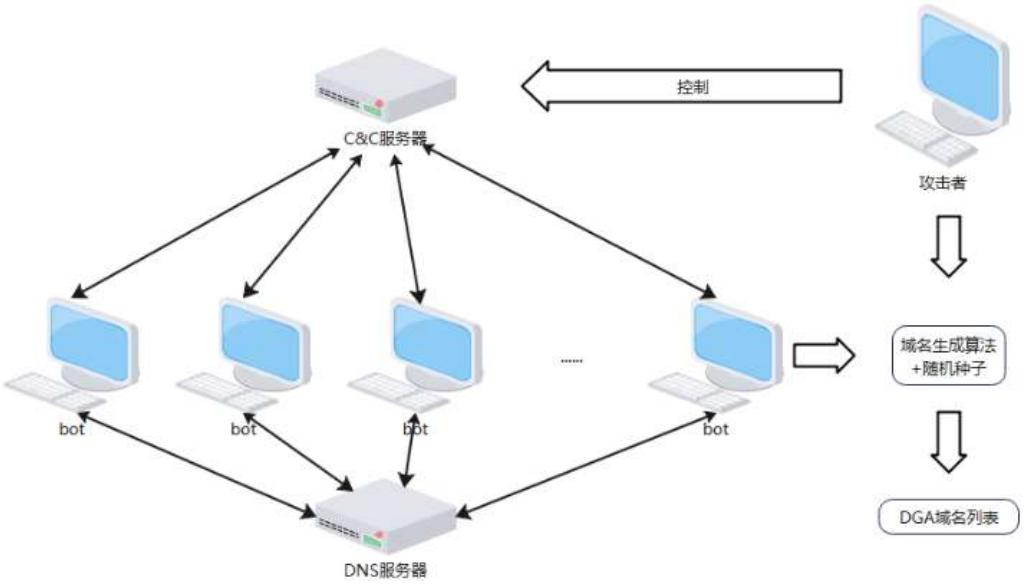

图 2-1 DGA 攻击模型

Figure 2-1 DGA Attack Model

僵尸网络是网络犯罪的一种复杂形式，其隐蔽性和破坏性使得防范和打击都极具挑战性。随着互联网技术的发展，僵尸网络也不断进化，采用更加高级的技术来逃避检测，面对更复杂的DGA算法，需要研制对应场景的检测机制，才可以应对僵尸网络带来的威胁。

2.2 DNS 域名系统

域名系统（DNS）是互联网的基础服务，负责将人类可读的域名（如 www.example.com）转换为机器可读的 IP 地址（如 192.0.2.1）。这一转换过程通过一个分布式数据库实现，该数据库包含了所有有效域名的映射信息[26]。DNS 的引入让互联网的域名空间可以动态地扩展，并且可以通过分布式数据库进行管理。

DNS 的工作原理包括域名解析和域名注册。域名解析过程首先在用户的本地 DNS 缓存中查找是否有相应的记录。如果没有，则查询用户的 ISP（互联网服务提供商）提供的 DNS 服务器。如果该服务器也没有找到记录，查询会继续传递到更高级别的 DNS 服务器，最终到达根域名服务器[27]。根域名服务器会指派到负责管理相应顶级域（如.com、.org）的域名服务器，然后逐级找到负责管理二级域名（如 example.com）的域名服务器，最终获得该域名的 IP 地址。域名注册需要使用 DNS，首先需要注册一个域名一般通过域名注册商完成，注册商会在 DNS 的层级结构中创建条目。

DNS 采用了一种分层的命名空间结构，从顶层开始，分别是根域名服务器、顶级域名服务器和权威域名服务器。根域名服务器负责管理顶级域名（TLDs），如.com、.org 等。顶级域名服务器管理二级域名，如.com 和.org 下面的所有三级域名。权威域名服务器管理具体的域名，如 example.com。域名解析过程如图 2-2 所示。

随着互联网的发展，DNS 的安全性和隐私保护变得越来越重要。DNS 查询和响应在互联网上是公开的，这意味着它们可以被中间人攻击者拦截和篡改。为了提高安全性，DNS over HTTPS（DoH）和 DNS over TLS（DoT）等技术被开发出来，以确保 DNS 查询和响应在传输过程中加密，防止中间人攻击。此外，DNS 隐私也很重要。由于 DNS 查询记录可以揭示用户的网络活动，一些隐私保护措施可以保护用户的隐私。如使用加密的 DNS 服务（如 Cloudflare 的 1.1.1.1）和本地 DNS 代理。

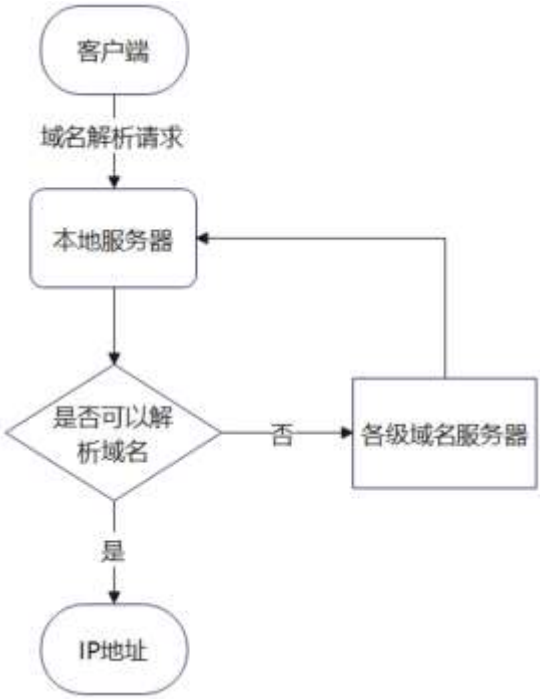

图 2-2 域名解析过程示意图

Figure 2-2 Schematic diagram of domain name resolution process

2.3 DGA 恶意域名

2.3.1 DGA 概述

DGA（Domain Generation Algorithm）是一种在恶意软件中广泛使用的技术，它基于随机字符，生成大量的域名，以便于恶意软件在命令和控制（C&C）服务器之间进行通信。这种技术的出现和发展与网络安全领域的发展密切相关。

DGA（域名生成算法）的历史可以追溯到 2007 年，当时由 Paul Royal 和 J. Alex Halderman 等人首次提出。他们的研究揭示了恶意软件中使用 DGA 的可能性，并指出这种技术可以提高恶意软件的隐蔽性和逃避安全检测的能力。这一发现标志着网络安全领域的一个重大突破，因为DGA技术的使用使得恶意软件能够以更隐蔽和难以追踪的方式进行操作。

随着互联网的普及和网络攻击的增加，DGA 技术在恶意软件中得到了广泛应用。一些著名的恶意软件，如 Conficker、Zeus 和 Mirai 等，都使用了 DGA 技术来生成和更新它们的 C&C 域名。这些恶意软件通过 DGA 技术能够灵活地更改它们的 C&C 域名的 IP 地址，从而逃避安全检测和阻断。这种动态域名生成的能力使

得恶意软件能够有效地规避传统的基于黑名单的安全防御措施，因为黑名单策略通常依赖于预先已知的不良域名或 IP 地址[28]。DGA 技术的广泛应用也促使了网络安全领域的研究和发展。为了应对DGA带来的挑战，研究人员和网络安全专家开始开发各种技术和工具来检测和阻止 DGA 生成的域名。这些技术和工具包括 DNS 监控、黑名单、启发式分析和机器学习等。随着这些技术和工具的发展，网络安全领域对 DGA 技术的应对能力得到了提高。然而，随着 DGA 技术的不断发展和改进，恶意软件制作者也采用了更复杂的策略来生成域名，使得检测和阻止 DGA 生成的域名变得更加困难。这要求研究人员和网络安全专家不断更新和改进他们的技术和工具，以应对新的挑战。DGA 技术的发展历史是与网络安全领域的发展紧密相连的，随着网络攻击的不断增加，DGA 技术在恶意软件中的应用也日益广泛。然而，随着检测和阻止 DGA 技术的方法的发展，网络安全领域对 DGA 技术的应对能力也在不断提高。

为了应对DGA技术带来的挑战，研究人员和网络安全专家开始开发各种技术和工具来检测和阻止 DGA 生成的域名。这些技术和工具包括 DNS 监控、黑名单、启发式分析和机器学习等。DNS 监控技术允许安全人员实时跟踪和分析域名解析活动，从而发现和阻止 DGA 生成的恶意域名。通过监控 DNS 请求和响应，安全系统可以识别出不寻常的域名查询模式，这些模式可能表明恶意软件的存在。黑名单是一种传统的网络安全策略，它包含已知的不良域名或 IP 地址[29]。当安全系统检测到与黑名单中的条目匹配的域名或 IP 地址时，它会自动阻止或拦截相关的网络流量。虽然黑名单对于已知威胁非常有效，但它们无法应对DGA生成的动态域名。启发式分析是一种基于规则的检测方法，它使用预定义的规则来识别可疑的域名生成模式。这种方法的关键在于能够识别出那些尽管不在黑名单中，但仍然表现出恶意行为的域名。然而，随着DGA技术的不断发展和改进，恶意软件制作者也采用了更复杂的策略来生成域名，使得检测和阻止DGA生成的域名变得更加困难。这要求研究人员和网络安全专家不断更新和改进他们的技术和工具，以应对新的挑战。

在本篇文章中，DGA 恶意域名是应用在僵尸网络中，通过指定算法生成随机的域名。在僵尸网络中被控制的主机也成为肉鸡，攻击者通过操纵肉鸡，在随机生成的域名列表中选取多个在 C&C 服务器上进行注册，由此便可建立肉鸡与 C&C 服务器之间的连接。建立连接后，可以利用 C&C 服务器对肉鸡下发命令，进行各种恶意网络行为。相反，合法域名则是通过正规域名注册机构注册，并且其目的是为了进行合法的网络活动。总的来说，DGA 技术的发展历史是与网络安全领域的发展紧密相连的。随着网络攻击的不断增加，DGA 技术在恶意软件中的应用也日益广泛。然而，随着检测和阻止DGA技术的方法的发展，网络安全领域

对 DGA 技术的应对能力也在不断提高。下表 2-1 中列出了一些典型的 DGA 域名家族及其对应的域名样例。

表 2-1 DGA 域名家族及域名样例

Table 2-1 DGA Domain Family and Domain Name Examples

恶意域名家族	域名实例
Emotet	wsbqbkfbytmhlwqx.en
	bluetiqaytgjdewf,eu
Ramnit	gjvoemfgcb.com
	vqnhnjgfrhkbouyvecf.com
Gameover	lyjm9lfgdrhr9m2bpdgrbsapzj.net
	lgnrdry16g64ali7mxrewcmjbnh.org
Simda	puerteg.net
	bofjjhef.net
Pykspa2	suwfogfoya.info
	hnorgffox.org
Virut	noiuik.com
	mttrgh.com

2.3.2 DGA 生成算法

在域名生成算法（DGA）中，主要由种子和算法两部分组成，种子（seeds）是用于生成域名的初始值或参数。可以根据种子和算法性质的不同，对 DGA 恶意域名生成算法进行分类。

基于种子性质的不同，可以根据是否依赖时间分为四类：TID（Time-Independent and Deterministic），这类种子不依赖于时间，且其值是可确定的。这意味着无论何时运行，这些种子都会产生相同的值[30]。例如，一个硬编码在恶意软件中的常量字符串可以被视为 TID 种子；TDD（Time-Dependent and Deterministic），这类种子依赖于时间，但它们产生的值是可确定的。例如，使用当前日期或时间的某些固定组合（如年份、月份、日期）作为种子，虽然这些种子会随时间变化，但每次运行时都会产生相同的域名模式；TDN（Time-Dependent and Non-Deterministic），这类种子依赖于时间，且它们产生的值是不可确定的。例

如，使用当前日期和时间的某些随机组合（如年份、月份、日期的随机排列）作为种子，每次运行时都可能产生不同的域名模式；TIN（Time-Independent and Non-Deterministic），这类种子不依赖于时间，且它们产生的值是不可确定的。例如，使用随机数生成器生成的随机字符串可以被视为 TIN 种子，这些种子在每次运行时都会产生不同的值，且与时间无关。

基于生成的算法的不同，也可以分为四类[31]：基于算术的种子是指使用简单的算术运算来生成域名的算法。这种算法通常涉及对种子值（可能包括常量、变量或两者结合）进行加法、减法、乘法、除法等操作，然后将结果转换为字符串形式，以生成域名。这种算法最终会生成一组可用 ASCII 编码表示的值，从而构成DGA域名，此类方法较为常见。例如，恶意软件可能会使用简单的加法、乘法或其他算术运算来组合种子值和固定字符串，以生成域名；基于哈希的种子使用哈希函数来生成域名的算法。哈希函数是一种将输入数据（种子值）转换为固定长度输出值（哈希值）的函数，输出值通常是一个较小的数字或字符串[32]。在 DGA 中，哈希函数用于生成难以预测的域名，因为每次输入相同的种子值时，哈希函数都会产生不同的输出值。例如，恶意软件可能会将种子值作为哈希函数的输入，并使用哈希结果的一部分来构建域名[33]。哈希函数通常会产生固定长度的输出，恶意软件可以选择输出的特定部分来使用。基于字典的种子，这类种子使用预定义的字典来生成域名。这种方法通常涉及从一组预先定义的字符中选择字符或字符组合来构建域名。字典可以是任何包含有效字符的集合，这些字符可以是字母、数字、特殊字符等。例如，恶意软件可能会从一个包含已知有效字符的字典中随机选择字符，或者根据某种规则组合这些字符来生成域名。组合种子：这类种子结合了上述多种方法来生成域名。例如，恶意软件可能会使用基于算术的种子来确定域名的一部分，同时使用基于哈希的种子来确定另一部分，或者结合使用基于字典的种子来生成整个域名，此类种子生成方法有效提高了域名的随机性和不可预测性[34]。

从上述内容可以了解到，DGA 生成的域名具有动态性和不可预测性，可以从多个方面评估和比较不同DGA算法的逃避检测能力，包括随机性和不可预测性、域名注册和解析的频率、种子和算法的复杂性、对抗性测试、实际恶意软件案例、资源消耗以及更新和适应能力。通过综合这些指标和方法，可以对DGA算法的逃避检测能力进行画像。但由于恶意软件和DGA算法不断演变，这种评估是一个动态过程，需要持续监控和更新。

2.4 机器学习算法

机器学习是人工智能（AI）的一个分支，它使计算机系统能够从数据中学习并做出预测或决策，而无需显式编程。机器学习的核心是创建算法模型，这些模型能够从数据中自动发现模式和规律，并利用这些模式和规律进行预测或决策。

机器学习被广泛应用于各个领域，包括金融、医疗、推荐系统、自然语言处理等。比如在金融领域中，通过分析历史交易数据，机器学习模型可以预测股票价格、评估贷款风险或识别欺诈行为；而在医疗健康中，机器学习可以帮助诊断疾病、预测疾病的发展趋势，甚至辅助医生进行治疗决策。机器学习也可以完成各种推荐系统，比如通过分析用户的历史行为和偏好，机器学习模型可以推荐商品、音乐、电影等。在自动驾驶方面，器学习算法可以处理来自车载传感器的数据，以实现自动驾驶汽车的目标检测和路径规划。本文主要使用机器学习算法完成网络中的域名分类，而在自然语言处理方面，机器学习模型可以用于文本分类、情感分析、机器翻译等任务。随着大数据和计算能力的提高，机器学习技术在未来的应用前景将更加广阔。

机器学习算法包括线性回归、逻辑回归、支持向量机、决策树、随机森林、神经网络等。这些算法可以应用于多种场景，如分类、回归、聚类、关联规则学习等。机器学习主要包括监督学习、无监督学习和强化学习三种类型，它们各自适用于不同的数据分析和预测问题。监督学习是一种学习算法，它使用标记的数据进行训练，以预测新的输入数据。监督学习算法包括线性回归、逻辑回归、支持向量机等。无监督学习则不使用标记的数据，而是通过发现数据中的模式和关系来对未标记的数据进行聚类、降维和特征提取。无监督学习算法包括 K-means、PCA、t-SNE 等。无监督学习的目标是发现数据中的内在结构和关系，以便更好地理解和分析数据。强化学习是一种通过与环境的交互来学习算法，它通过试错的方式学习做出最优决策。强化学习算法包括Q学习、SARSA和深度Q网络等。强化学习的目标是通过与环境的交互，学习做出使累积奖励最大化的决策。这三种学习方法各有特点和适用场景。监督学习适用于预测和分类问题，无监督学习适用于发现数据中的模式和结构，强化学习适用于决策和控制问题。本文的 DGA 恶意域名检测算法主要采用无监督学习算法对域名进行聚类，在使用自监督学习针对聚类后的域名进行分类，在后续章节中会详细介绍采用的具体算法

在实际应用中，机器学习算法的实现包含几个主要步骤。首先，数据收集阶段是基础，一般会从各种渠道获取相关的数据集，这些数据集既包括用于训练模型的样本，也包含用于测试模型性能的独立数据。接下来，是针对数据的预处理操作，它涉及到对原始数据进行清洗，去除噪声和异常值，以及进行归一化和编码等操作，以确保数据质量，作为后续任务的输入。

在特征选择阶段，需要从大量潜在特征中挑选出对预测任务最相关的特征子

集，这可以显著提高模型的效率和准确性。模型选择则是根据具体问题选择合适的算法，这可能涉及到比较不同算法的性能，如决策树、随机森林、神经网络等。在训练模型阶段，使用训练集对选定的模型进行训练，调整模型参数，优化模型性能，这是一个迭代的过程，可能需要多次调整和验证。

模型的评估是衡量其性能的关键步骤，通过在测试集上应用模型，可以得到模型的准确率、召回率、F1 分数等指标，从而全面评估模型的实际应用能力。根据评估结果，模型调优阶段会对模型参数和结构进行调整，由此可以进一步提高模型的性能和泛化能力。

最终，将训练好的模型部署到实际应用中，进行预测或决策，这是机器学习项目的最终目标。模型部署需要考虑算法的实时性、效率和稳定性，确保模型能够在实际的目标环境中稳定运行，提供准确的预测或决策支持。

2.4.1 聚类算法

在机器学习和数据挖掘领域，聚类算法是一种关键的无监督学习方法，其基本功能是将一组数据划分成若干个群组，这些群组通常被称为“簇”。在这些簇内部，数据对象之间具有较高的相似性，而不同簇的数据对象则表现出较低的相似度。聚类算法的核心目标在于揭示数据的内在结构，将相似的数据归为一类，而将不相似的数据分到不同的类中。

作为一种无监督学习方法，聚类算法的一个显著特点是其不需要预先标记数据，这意味着算法可以在没有任何预置知识的情况下，独立地从数据中挖掘出潜在的模式和结构。聚类算法具有自组织性，能够自动地识别并划分数据，使得相似的数据紧密地聚集在一起，而不同簇的数据则保持一定的距离。

通过聚类算法，可以更好地理解数据的内在性质和关联，从而深入地分析和解读数据。在许多实际应用中，如数据挖掘、图像分割、社交网络分析等领域，聚类算法都发挥了关键作用，帮助本文发现数据中的隐藏结构，揭示数据之间的关系，以及提取数据中的重要特征。

本文采用凝聚的层次聚类实现DGA恶意域名检测中的域名聚类，层次聚类是一种将数据分组为嵌套簇的聚类方法，形成一个层次结构。在层次聚类中，数据对象通过距离度量（如欧式距离、曼哈顿距离或余弦相似性）进行相似性评估。该方法的基本思想是，每个数据点最初作为一个单独的簇，然后逐渐合并成相似的簇。

层次聚类可以进一步分为凝聚的层次聚类和分裂的层次聚类。凝聚的层次聚类和分裂的层次聚类。在凝聚的层次聚类中，每个数据点开始时都是一个单独的

簇，然后逐渐合并最相似的两个簇。这个过程重复进行，直到所有的数据点都合并到一个大簇中，或者满足某个预设的停止条件（如预设的簇数或簇间距离不再显著变化）。在分裂的层次聚类中，所有数据点开始时合并到一个簇中，然后逐渐将一个簇分裂成两个更小的簇，如果这样做能够降低簇内数据点的平均距离。这个过程也重复进行，直到满足某个停止条件。最终结果形成了一个嵌套的簇层次结构，有助于理解数据对象之间的层次关系。层次聚类特别适合于探索性数据分析，因为它允许可视化和调整聚类的层次结构，从而更好地理解数据。

2.4.2 决策树

在监督学习领域，决策树算法因其强大的数据建模和预测能力而应用广泛，包括应用于分类与回归任务等。该算法的核心在于通过一系列逻辑判断，将输入数据集划分成多个子集，每个子集满足特定的划分条件。这一过程通过树形结构实现，其中每个节点表示一个特征及其对应的判断条件，而叶节点则代表了最终的分类结果或预测值。

决策树算法的显著特点包括其自顶向下的递归划分策略，这一策略有效地将数据集拆分成更小的、满足特定条件的子集。此外，决策树在构建过程中自动选择最优特征进行划分，这一选择过程基于信息增益、Gini 指数或基尼不纯度等标准，确保了划分效率和准确性。为了防止过拟合现象，决策树算法通常采用预剪枝或后剪枝技术，以去除不重要的分支，增强模型的泛化能力。

决策树的可解释性也是其一大优势，其树形结构清晰直观，每个节点的判断条件都与数据集中的特征和阈值相对应，使得模型决策过程易于理解和分析。此外，决策树能够捕捉数据中的复杂模式和关系，展现出良好的泛化能力。然而，决策树算法也存在一些局限性，如对异常值的敏感性和过拟合的风险。

为了解决这些问题，提出了一系列决策树算法的改进版本，如随机森林和 XGBoost 等。这些算法通过集成多个决策树，引入正则化技术或优化剪枝策略，有效地提高了模型的稳定性和预测性能。

综上所述，决策树作为一种高效且实用的监督学习算法，不仅能够快速学习数据中的有用模式和规律，还能提供准确的预测和决策支持。尽管存在一些局限性，但通过调整超参数和引入先进的集成技术，决策树算法在众多领域都展现出了其独特的优势和应用潜力。

2.4.3 随机森林生成算法

随机森林（Random Forest，简称 RF）是一种集成学习方法[35]，通过构建多棵决策树并集成它们的预测结果来提高模型的预测性能。该算法由 Leo Breiman 于 2001 年提出，结合了 Bagging 集成学习理论和随机子空间方法[36]。每一棵决策树都是基于随机采样的数据集训练的，这些数据集包含了原始数据的一个子集和特征的一个子集。

在介绍随机森林之前，需要先了解 Bagging，Bagging（Bootstrap Aggregating）[37]是一种集成学习技术，它可以在提高机器学习模型的泛化能力的同时减少预测的方差，从而增强预测的准确性。它是通过构建多个独立的训练模型并组合它们的预测结果来实现这一目标的。Bagging 的核心思想是创建多个训练数据集，每个数据集都是通过对原始数据集进行自助采样（bootstrap sampling）[38]得到的。自助采样是一种有放回的抽样方法，即每次从原始数据集中随机抽取一个样本，直到所有样本都被抽取过[39]。最后组合模型得出的推断可以是这些基学习器的平均值或者投票结果 [40]。从原始数据集中随机选择一个子集（bootstrap sample）进行训练，这个过程重复进行多次，每次训练一个决策树。每个决策树都是基于不同的子集训练的，这样可以减少个别决策树对随机噪声的敏感性。通过汇总这些预测结果，可以有效降低随机误差，从而增强集成模型的稳定性和推广能力。

Bagging 可以与多种类型的基学习器结合，如决策树、神经网络和支持向量机等，其优势在于能够在保持基学习器偏差不变的情况下，显著减少模型的方差，并在训练数据较少时仍能展现良好的泛化性能。Bagging 的流程如图 2-3 所示。

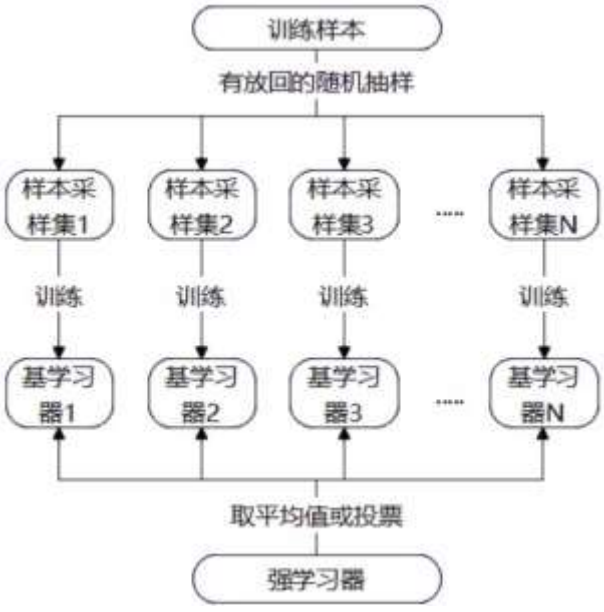

图 2-3 Bagging 流程

Figure 2-3 The Bagging process

随机森林（Random Forest）的随机性主要表现在两个方面：数据采样和特征选择。

首先，在数据采样方面，随机森林采用自助采样（Bootstrap Sampling）方法从原始数据集中随机抽取样本，创建多个训练数据集。每个训练数据集的大小与原始数据集相同，但包含的样本是不同的。自助采样也是一种有放回的抽样方法，即每次从原始数据集中随机抽取一个样本，直到所有样本都被抽取过。这种随机抽样方法使得每棵决策树训练的数据都是随机的，增加了模型的随机性[41]。

其次，在特征选择方面，随机森林在训练每棵决策树时，从所有特征中随机选择一部分特征进行分裂。这意味着每棵决策树在选择最优特征进行分裂时，都是基于一个随机子集进行的。这种随机特征选择方法进一步增加了模型的随机性。

通过引入数据采样和特征选择的随机性，随机森林能够在一定程度上减少模型的方差，提高模型的泛化能力和稳定性。这种随机性也是随机森林区别于其他集成学习方法（如 Bagging）的一个重要特点。

随机森林算法模型的目的是通过多棵树的投票或平均预测来减少模型方差，提高模型的稳定性和预测准确性。随机森林模型的形成过程如图 2-4 所示：

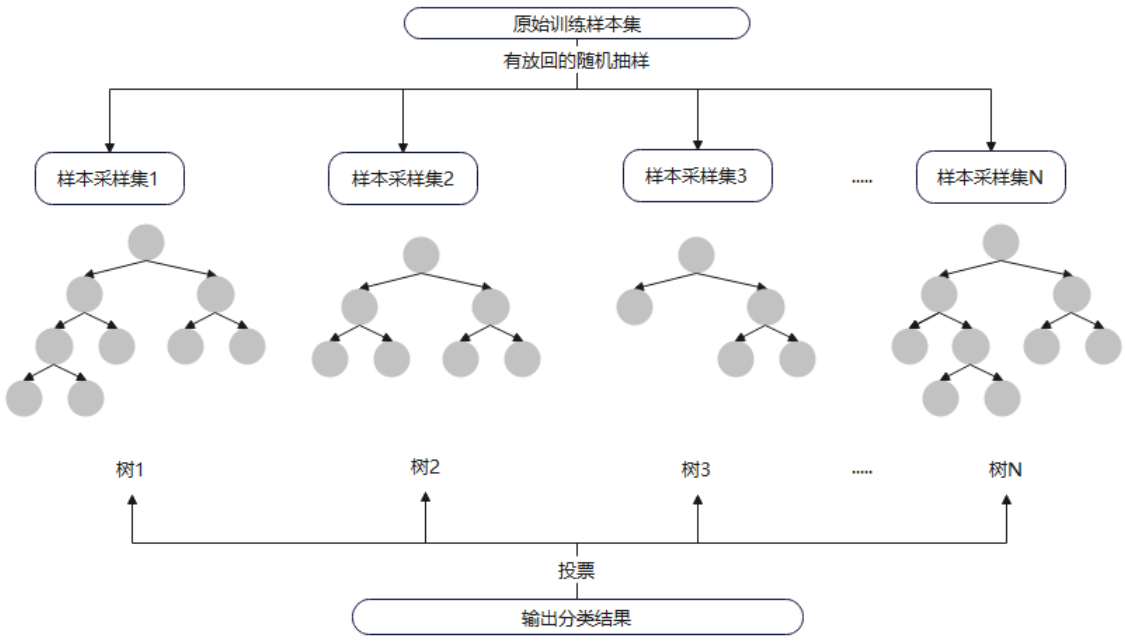

图 2-4 随机森林模型

Figure 2-4 The Stochastic forest model

根据模型生成图展示所示，随机森林模型的生成主要包含以下几个步骤：首先，从样本集中随机抽取一定数量的样本，并随即将这些样本放回，以便下一次

抽样时能够包含不同的样本。其次，从所有特征中随机选取一部分特征，使用这些随机选择的特征来训练决策树，通常采用分类回归树（Classification And Regression Tree）[42]。在完成这些步骤后，将得到一定数量的训练好的决策树，这些树基于不同的数据集和特征集训练，从而具有独特的特性和预测能力[43]。对于分类任务，随机森林采用简单多数表决法来确定新数据的类别，而对于回归任务，则采用简单平均法来汇总每棵树的输出结果。在构建每棵决策树时，确保数据的随机性和特征选择的随机性是至关重要的，以增强模型的泛化能力。

2.5 本章小结

本章首先介绍了僵尸网络的攻击方式、攻击原理及其在网络安全领域的威胁。着重介绍了 DGA（Domain Generation Algorithm）技术在僵尸网络中的关键作用。在理解 DGA 技术的基础上，本章进一步介绍了 DNS 域名系统的组成结构和解析原理。包括域名解析和域名注册的具体过程。通过对 DNS 系统的了解，本文能够更好地理解 DGA 技术在实际攻击中的应用。

接着，本章阐述了 DGA 相关理论知识，包括 DGA 的概述和生成算法。分析了不同类型的 DGA 算法，包括基于时间、基于算术和基于字典的 DGA 生成算法，并探讨了它们在实际攻击中的特点和应用。通过对 DGA 算法的深入分析，本文能够更好地理解 DGA 技术的工作原理和特点。

在本章的最后，对机器学习的常用算法进行了介绍，包括聚类算法、决策树以及随机森林生成算法。这些算法在网络安全领域中具有广泛的应用。通过对这些算法的了解，可以更好地应用机器学习技术来检测和评估 DGA 恶意域名。

这些内容为后续章节中进行 DGA 恶意域名检测评估的实验和实现 DGA 恶意域名检测与评估系统提供了理论基础。本研究将在以上内容的基础上，进一步探讨如何开发实现 DGA 恶意域名检测与评估系统，以提高网络安全领域的防范能力。

3 DGA 恶意域名检测方法

本章首先对现有的DGA恶意域名检测方法进行了细致分析，提出一套契合本系统目标使用场景的检测流程。通过对比DGA域名与合法域名的特征，本文确定了流程中所采用的特征和算法。经过验证，本检测流程展现出较好的性能，同时实现了低误报率，符合本系统的设计目标。

3.0 检测方法设计

在DGA恶意域名检测中，为了有效的区别恶意域名和合法域名，研究者提出过一系列检测流程。在使用深度学习的检测方法中，虽然检测精度普遍较好，但检测耗时长且存在过拟合、误报高的问题，此外还有基于域名字符特征的检测，采用传统机器学习聚合或分类算法[44]实现的 DGA 检测流程，这类方法仅仅基于域名字符的特征进行判断[45]，在大数据的场景下，存在相同的问题[46]。

本文研究的目标使用场景，主要指在大数据的背景下，对性能要求高且数据较为机密的企业，比如政务、医疗系统。此类系统需要稳定性好、比起高精度更需要低误报、可以快速检测的安全检测评估系统。并且这类系统一般无法使用云上环境，数据只能保留在内部网络中，需要以本地部署方式的安全检测系统。本文对比了上述已有的检测流程后，设计并实现一套适用于目标场景的DGA恶意域名检测系统。

在 DGA 检测方法研究初期，构想了一种先按照时间分组的 DGA 域名检测，这个方法首先要按照 DNS 请求的时间间隔对域名进行聚类或者分类操作，然后提取一些行为和字符特征，比如访问次数、域名长度等特征信息，基于这些特征进行后续的DGA检测。实验结果表明，这种基于组特征的检测方法主要的问题的会消耗大量的内存资源，且检测时间较长。比如在实验中，本文使用了具有 32 个 CPU 的服务器进行系统的搭建部署。但是在处理某个大数据环境下的流量及日志信息时，特征提取部分共消耗 15 小时，基本无法达到目标场景中实时检测 DGA 域名的需求，而且域名的分组方法也对结果有较大影响。实验结果表明，很难满足稳定的低误报快速检测，所以本文为了满足目标场景下的检测需求，本文考虑使用轻量级特征进行域名分类。但是，已有研究表明，在大数据环境下，仅依靠域名字符特征进行分类分析检测 DGA 域名存在较高的误检率。而 DGA 的域名请求在行为上存在一些规律。比如，请求的次数、请求的时间间隔等。基于上述这些实践和分析，采用以下思路设计并实现了DGA检测方法，主要包含以下三个方

面：

（1）整体检测分为两个阶段，第一个阶段使用聚类算法对具有相似性的DGA 域名进行聚类，以便发现和挖掘这些域名之间的内在联系和规律。第二阶段采用分类算法实现对域名的二分类操作。

（2）在聚类分析阶段，使用特征包括域名字符特征和访问次数等访问行为特征，有效降低 DGA 域名的误报率。

（3）在分类阶段，采用以域名字符特征为主的轻量级特征检测，满足目标场景中的大流量快速检测要求。

3.1 检测方法实现

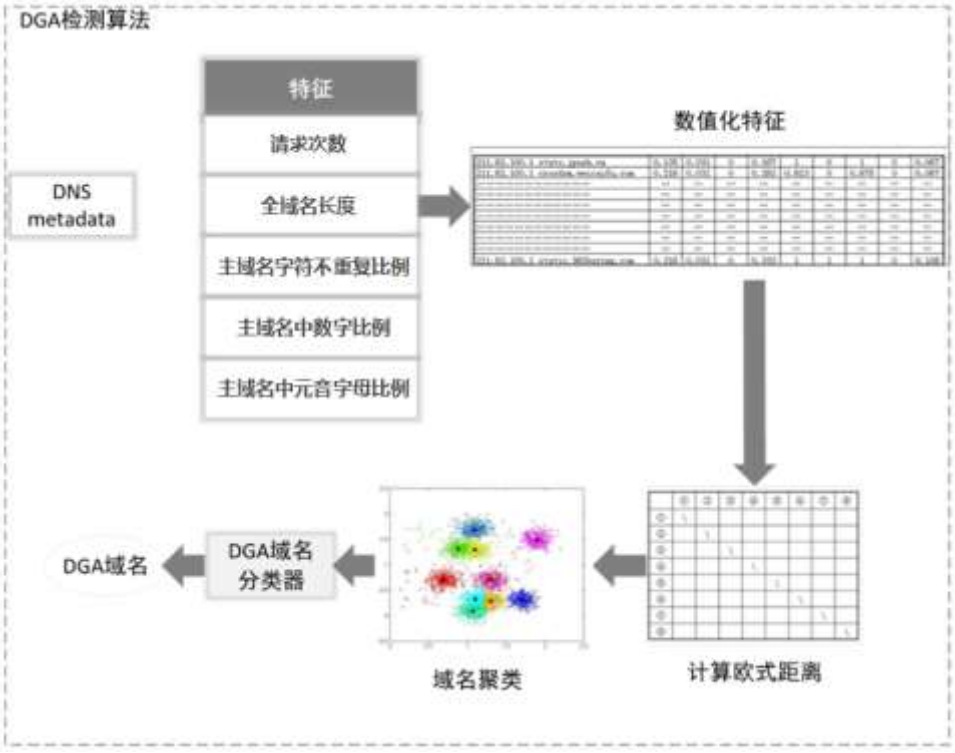

图 3-1 DGA 检测模型流程图

Figure 3-1 Flowchart of the DGA Detection Model

基于上文的检测流程设计思路，构建检测模型如图 3-1 所示。根据相同家族的DGA域名相似性和域名本身的随机性，进行有效识别和分类。在此过程中，首先利用聚类算法对具有相似性的DGA域名进行聚类，以便发现和挖掘这些域名之间的内在联系和规律。其次，考虑到DGA域名本身具有的随机性，本文采用随机森林分类算法对聚类结果进行进一步的判断和分类，以提高分类的准确性和稳定

性。核心步骤的具体实现包括以下四点：

（1）聚类特征选取及数值化

在模型聚类之前，需要对特征进行提取和处理，在深入探索域名属性表征后，从每个域名中提取了相关特征，其中包括域名字符的特征和访问行为的特征。特征的选取应该在DGA域名和合法域名上有明显区别。例如主域名中元音字母比例和主域名中数字比例在不同域名上的表现如图 3-2：

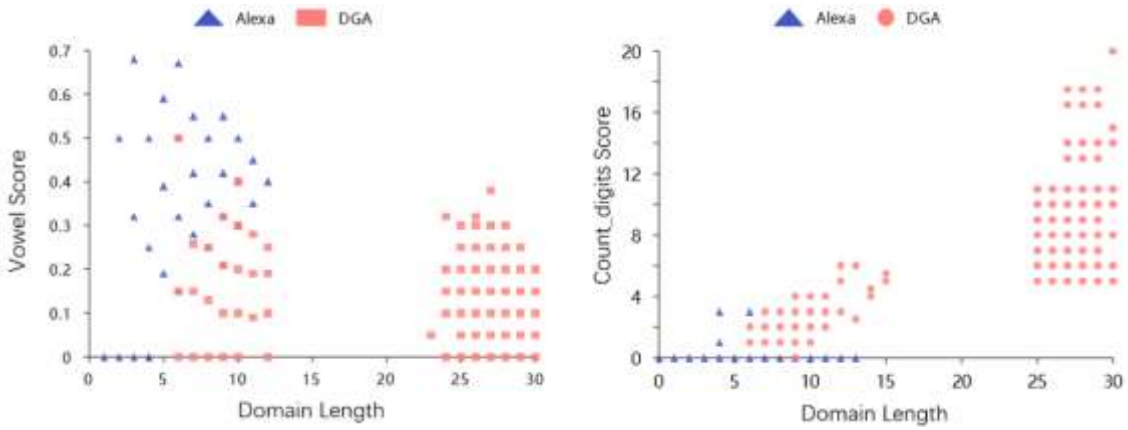

图 3-2 不同特征的表现

Figure 3-2 The Performance of Different Features

通过实验分析大量的域名数据，将这些特征与正常域名进行对比。确定了相关性较高的特征属性，如表 3-1 所展示的：

表 3-1 聚类特征表

Table 3-1 Clustering Feature Table

域名特征	#
请求次数	F1
全域名长度	F2
主域名中字符不重复比例	F3
主域名中数字比例	F4
主域名中元音字母比例	F5

为验证特征有效性，本文采纳了基于特征相关性分析的策略。此方法的核心在于通过皮尔逊相关系数的计算，对各个特征间的相互关系进行量化。经过细致的统计分析，所得结果表明所选取的特征集并未呈现出显著的多重共线性，这一

结论不仅说明了这些特征在统计学上的独立性，而且强调了它们各自携带的独到信息价值。因此，本文所采用的特征提取方法，不仅确保了每个特征在描述域名独特性质时能够贡献其独有的视角，同时也为后续的分析和模型构建奠定了坚实的数据基础。

提取特征后，在聚类之前还需要被归一化，归一化对于基于距离度量的算法非常重要，如 k-means、和凝聚聚类（Agglomerative clustering）。因为它可以使得不同特征的量纲一致，这些算法在聚类过程中依赖于距离度量来确定数据点之间的相似性或相异性。首先，特征归一化有助于消除不同特征间量纲和数值范围差异的影响，从而确保在计算数据点之间的相似性或相异性时，每个特征都能公平地参与。在没有归一化的情况下，数值较大的特征可能会在距离计算中占据主导地位，导致其他特征的影响被相对削弱，从而影响聚类结果的准确性。其次，特征归一化还能够提高聚类算法的效率。由于归一化减少了算法在优化过程中需要调整的参数数量，因此可以加快算法的收敛速度，尤其是在处理大规模数据集时。最后，正确的特征归一化有助于改善聚类质量。在凝聚聚类等算法中，如果没有进行归一化，可能会导致算法错误地将实际上不相似的点聚在一起，或者将相似的点分开，从而影响聚类结果的可靠性。

在本篇中，特征归一化是将所有的特征值归一至 0-1 的范围，算法中采用的是最小-最大缩放（Min-Max Scaling），归一化转换函数如下：

$$x' = (x - x_{min})/(x_{max} - x_{min}) \qquad （式 3-1）$$

其中 x 是原始特征值，x_{min}和x_{max}分别是特征的最小值和最大值。这种方法将特征缩放到 [0, 1] 区间内。最小-最大缩放是一种简单的归一化方法，不会引入额外的计算复杂性。这种特点使得此方法非常适合本篇系统应用的特定场景，有助于保持算法的效率，尤其是在处理大规模数据集时。

（2）域名聚类

通过上文的介绍，基于相同家族的DGA域名相似性特点，利用聚类算法对具有相似性的 DGA 域名进行聚类，以便发现和挖掘这些域名之间的内在联系和规律，在本研究中使用凝聚聚类的层次聚类方法，凝聚聚类是一种自下而上的方法，它从单个数据点开始，逐渐合并相似的数据点，形成聚类。这种方法有助于揭示数据中的自然分组，对于检测DGA生成的域名来说，可以更好地识别可能由同一算法生成的域名簇。聚类算法在检测DGA生成的域名时提供了一种灵活、自适应的方法，能够更好地利用DGA域名中相同家族所具有的相似性特点，从而提高检测的准确性和效率。

在聚类过程中，本文应用了凝聚的层次聚类算法对基于欧氏距离的计算结果进行聚类。层次聚类是一种通过递归聚合数据集来形成越来越大的子集的方法。需要针对构造的特征数据矩阵，采用欧氏距离公式来计算数据矩阵中各个条目之间的距离，并基于二叉树的数据结构来构建层次聚类。欧氏距离是一种广泛应用于多维空间中两点间距离度量的数学公式。假设选择两个特征向量点 v 和 w，欧式距离公式如下所示：

$$d = [\sum_{i=1}^{n}(v_i - w_i)^2]^{\frac{1}{2}} \qquad （式 3-2）$$

其中，v_i和 w_i分别是向量 v 和 w 在第 i 个维度上的坐标。

通过计算欧式距离后在矩阵中寻找距离的最小值，并分析此最小值是否小于设定阈值，若小于设定的阈值则认为是同类域名，对于判定为同类的叶子节点执行合并操作，递归以上操作，直至满足最小值大于阈值或者所有域名聚合为一类收敛后终止。对 DGA 域名检测的聚类过程如下图 3-3 所示：

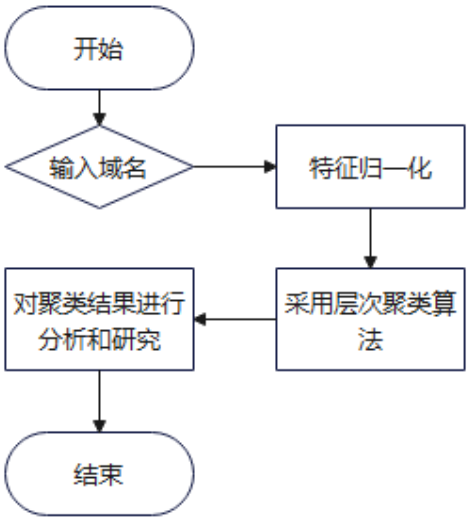

图 3-3 DGA 域名检测的聚类过程

Figure 3-3 Clustering Process for DGA Domain Detection

（3）分类特征提取及数值化

聚类之后针对每一个集合，进行域名的二分类操作，基于已有数据集，将特征数值化作为随机森林的输入，经过域名分类器的判断后，对聚类得到的多类计算标签并上报最终检测结果。

通过对恶意域名与常规域名之间的差异进行深入分析，将域名字符特征和域名与白样本相似性加权值作为分类器输入特征。针对黑、白样本，对域名本身的

字符特点，提取了域名熵、域名长度、Alexa域名、Tranco域名、英语单词及汉语拼音字符特征，如表 3-2:

表 3-2 分类特征表

Table 3-2 Clustering Feature Table

域名特征	#
域名长度	F6
域名字符串熵	F7
Alexa 3/4/5-gram 命中次数加权值	F8
英文单词表 3/4/5-gram 命中次数加权值	F9
汉语拼音 3/4/5-gram 命中次数加权值	F10

由于域名属于字符类的表示，与处理图像等数值信号的模型相比，存在显著差异。自然语言数据，也称为语料，由语言的基本单位组成，如英文中的单词或中文中的字。这些数据以字符串形式存在，而非数值形式。因此，在对特征进行计算之前，需要将这些字符串数据编码为可以处理的数值形式，经过转换后，将用特征向量输入随机森林模型中，这一步骤通常涉及将分类特征编码为二进制形式或其他数值表示方法。

在本研究中，采用了 N-Gram 方法对域名进行特征提取，使用转换器将域名字符串列表转换为 N-Gram 向量表示，并计算域名与白样本相似性的加权值。N-Gram 是一种基于 n 个连续字符出现概率来推断语句结构的统计语言模型。在此模型中，文本内容被划分为长度为 n 的字节片段，形成序列 Gram。随后，根据预设的阈值对这些 Gram 进行过滤，并统计其在文本中的出现频度，从而生成 Gram 列表。

选择 N-Gram 模型进行处理的原因在于其独特的优势。相较于其他算法，N-Gram 模型无需对文本内容进行复杂的语言学处理，从而降低了算法对语言的依赖性。此外，N-Gram 模型无需词典和规则的辅助，使其能够同时处理中英文文本。基于这些优势，本研究最终决定采用 N-Gram 模型对域名进行特征提取。经过处理，将所有域名的特征值整合成一个矩阵，其中每一行代表一个域名，每一列代表一个特征维度。通过采用 N-Gram 模型，本研究成功地将域名转换为具有丰富信息的特征向量，为进一步的机器学习任务奠定了基础。实验结果表明，基于 N-Gram 特征的域名处理方法在各项指标上均取得了优异的表现，验证了该方法的有效性。

（4）域名分类

在完成特征提取后，选择适当的分类器构建分类模型，其训练过程对实验结果具有显著影响。本研究旨在实现二分类任务，即区分黑白域名，以解决二分类问题。在机器学习算法中，常见的选择包括逻辑回归、决策树、随机森林、朴素贝叶斯等。针对常见的四种机器学习分类算法，在训练集上，采用了十折交叉验证（10-fold cross-validation）的方法来训练分类器模型。这一方法通过将数据集分割成十个子集，轮流使用其中一个子集作为测试集，其余九个子集作为训练集，以评估模型的性能。通过这种方法，本文得到了不同算法在训练集上的 ROC 曲线，如图 3-4 所示：

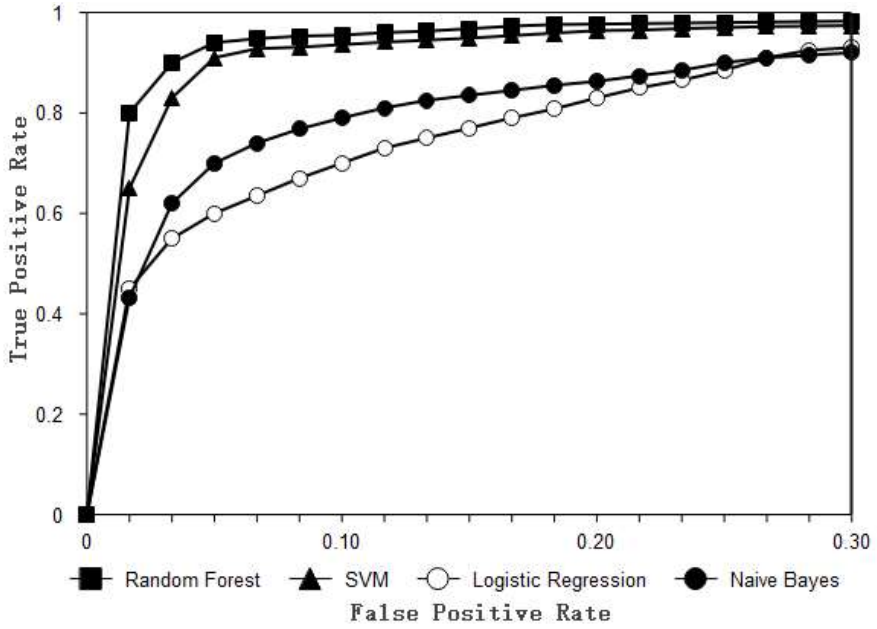

图 3-4 分类器性能比较

Figure 3-4 Classifier Performance Comparison

通过上图可知，本文的分类任务属于非线性分类任务，并且特征之间存在一些相关性，通过对比，随机森林算法的分类性能最优，鉴于随机森林算法的诸多优势，包括易于并行处理、实现效率高以及对数据噪声的敏感度低，本研究选择该算法作为检测模型分类的训练方法。

随机森林分类算法，通过集成多个决策树来实现分类，其中训练过程包括有放回的样本选择，确保每次生成决策树的随机性，从而有效预防过拟合问题。随机森林分类算法如图 3-5 所示，基于决策树算法构建，首先，在具有 N 条记录的数据集中随机有放回的抽取 i 条记录。然后，为每个样本构建一颗决策树，并在构建过程中，采用“最大深度”限制决策树的深度，以避免过拟合。最后，针对每颗决策树产生的输出，根据多数投票或对分类和回归进行平均来确定最终输出。

通过这种方法，随机森林算法能够有效地提高预测准确性，同时具有较好的鲁棒性。实验结果表明，随机森林算法在许多实际应用中均取得了优异的性能。

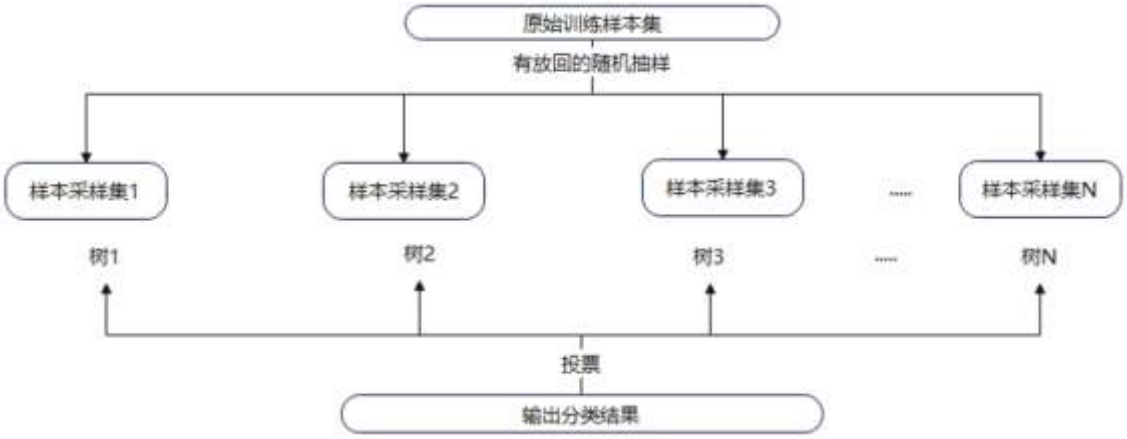

图 3-5 随机森林流程图

Figure 3-5 Random Forest Flowchart

通过上述的检测模型，可以在现网环境中有效地识别 DGA 恶意域名，提高检测的鲁棒性和效率。此外，还对算法进行了多次迭代和优化，以进一步提高其性能和鲁棒性。实验结果表明，该算法在DGA恶意域名检测方面具有较高的效率和较低的误报率，适用于对稳定性要求高的场景，有效地保护用户的网络安全。

（5）DGA 域名判定

在上述工作完成之后，将进行检测方法的最后一步，DGA 恶意域名判定。首先，确保聚类的域名数量不少于 10 个，以保证样本量的充足性。其次，检查所有域名标签数是否相同，即“.”的数量加 1 是否相等，这有助于确保域名结构的统一性。接着，计算域名重合比，若小于设定的阈值 5%，则表明这些域名在结构上具有相似性。然后，统计无应答域名的占比，若大于 30%，则可能存在大量的无效或无法访问的域名。最后，计算被DGA域名分类器检测出的占比，若大于 40%，则表明这些域名被分类器的识别准确度较高。只有满足以上所有条件的每类，才被检测模型认为是真正的 DGA 域名。

3.2 实验验证

3.2.1 实验环境

本实验所使用的硬件配置参数及软件的版本信息如表 3-3 所示：

表 3-3 实验环境说明

Table 3-3 Experimental Environment Description

环境配置项	详情
CPU	Intel(R) Core(TM) i7-9700K@ 3.60GHz
内存	16GB DDR4
操作系统	Linux
开发语言	Python 3.8

3.2.2 数据集

实验数据集主要包括四个部分，正常合法域名、DGA 恶意域名、英语单词和汉语拼音，在正常域名部分中，本文选取了来自 Tranco 排名前 25 万的白名单域名，这一部分的数据选择基于 Tranco 列表（访问网址：https://tranco-list.eu/list/8KYV/1000000）。相较于先前研究主要采用的 Alexa 列表[47](访问网址：http://alexa.com/topsites/），后者虽然广泛应用于识别良性域名，但有研究表明，Alexa 排名前 3 万的域名中也可能包含恶意域名[48]，且在 2022 年 Alexa 已停止了提供 top 域名列表。鉴于此，本研究的白名单域名采用了 Alexa 和每日更新的流行域名排行列表 Tranco 作为良性域名的数据来源。恶意域名数据集由两部分组成，一部分来源于 360 网络安全实验室提供[49]，在本次实验中，共收集了来自 11 个不同 DGA 家族的域名，域名种类丰富，可以有效覆盖当前存在的 DGA 恶意域名形式，包含 1,569,862 个黑名单域名。考虑到域名生成算法的随机性，另一部分主要来源于自主通过 43 个 DGA 算法生成的 2,418,903 域名，共计 3,988,765 个恶意域名。英语单词和汉语拼音部分主要描述了字符之间的关系。在数据集中，所有恶意域名域名均被视为 DGA 检测二分类问题中的阳性，其输出标签为 1。而对于 Alexa 数据集中的域名，则被视为 DGA 检测二分类问题中的阴性，其输出标签为 0。数据集详情如表 3-4 所示。

表 3-4 数据集分类

Table 3-4 Dataset Classification

类型	Alexa 域名	Tranco	DGA 域名	英语单词	汉语拼音
数量	750,000	250,000	3,988,765	364,738	183,294

在使用数据集之前，要对数据集进行预处理，去除数据中的重复数据、空数据，并且需要过滤数据集中的短域名，并且采用和文献[50]类似的处理方法，去除顶级域名。将 Alexa 域名及 Tranco 域名数据保存在 CSV 文件中，首先读取该文件，得到的域名信息如下表 3-5 所示：

表 3-5 合法域名

Table 3-5 Legitimate Domain Names

Top	URL
1	google.com
2	facebook.com
3	youtube.com
4	microsoft.com
5	netflix.com
6	twitter.com
7	tmall.com
8	instagram.com
9	qq.com
10	windowsupdate.com

在表中，"Top"列展示了域名在真实世界中的排名顺序，而"URL"列则记录了相应的网站域名。在准备数据集之前，必须对原始数据集进行一系列预处理步骤。首先，从"URL"列中提取域名部分，去掉顶级域名，并将它们复制到新创建的"Domain"列中,最终仅保存 "Domain"列即可，并对数据进行了清洗，最终获得了包含 876,632 条记录的数据集。在此之后，新增了"Tag"标识列，这一列用于标识数据集的类别。该列的值被设置为"0"，用以表示域名属于合法类别。完成这些步骤后，处理后的数据如表 3-6 所示。

对于异常数据集的处理，同样采用类似的步骤进行数据清洗和结构调整。首先去除恶意域名数据中的重复数据、空数据，然后，从剩余的原始数据中提取域名，删除顶级域名，并将它们移至新创建的"Domain"字段。并且最终仅保留此字段。在这一过程中，对数据集进行了净化。最后，添加了新的标识类别的字段"Tag"，其值被设定为"1"，以标识这些数据属于异常类别。处理后，总共得到了3,547,988 条记录,并存储在 CSV 文件中。预处理后的数据集如表 3-7 所示。

表 3-6 预处理后域名

Table 3-6 Preprocessed Domain Names

Domain	Tag
google	0
facebook	0
youtube	0
microsoft	0
netflix	0
twitter	0
tmall	0
instagram	0
qq	0
windowsupdate	0

表 3-7 DGA 域名

Table 3-7 DGA Domains

Domain	Tag
bbc16e2659b9b9b5128c2f7e5877d29b	1
f62b550a0e5e4f234fdd30c927665c91	1
fe28753777	1
93b375dd6cd9f2704d613d1016dbe0f2	1
336c986a284e2b3bc0f69f949cb437cb	1
d84a6a7a28	1
40a43e61e56a5c218cf6c22aca27f7	1
afcc0c1f4b9fd590a61ba1c24b49b525	1
e3bea872ae	1

针对预处理后的数据集，需要过滤数据集中域名字符长度小于 5 的短域名。随机抽取 80%的数据作为训练集，其余 20%为测试集，并分别将"Tag"为 1 的恶意域名与"Tag"为 0 的合法域名混合，作为后续模型的输入。合并后的数据集结果展示如表 3-8 所示。

表 3-8 合并数据集

Table 3-8 Merged Dataset

Domain	Tag
bbc16e2659b9b9b5128c2f7e5877d29b	1
f62b550a0e5e4f234fdd30c927665c91	1
fe28753777	1
93b375dd6cd9f2704d613d1016dbe0f2	1
336c986a284e2b3bc0f69f949cb437cb	1
d84a6a7a28	1
40a43e61e56a5c218cf6c22aca27f7	1
afcc0c1f4b9fd590a61ba1c24b49b525	1
e3bea872ae	1
google	0
facebook	0
youtube	0
microsoft	0
netflix	0
twitter	0
tmall	0
instagram	0
windowsupdate	0

3.2.3 实验结果分析

在本研究中，为了全面评估恶意域名检测模型的性能，本文进行了两个维度的对比实验：特征组合与检测方法。这些实验旨在衡量模型的检测效果和性能开销。本章节重点在于特征的选择及二分类算法，依据对当前域名检测研究的分析，轻量级特征在机器学习中的应用展现出了优越的性能。在域名检测中，常用的机器学习算法包括逻辑回归、支持向量机（SVM）、朴素贝叶斯和随机森林。研究发现，随机森林分类器在检测效果上更为出色。为了确保检测方法的最佳效果，本文将提取的特征应用在四种不同的分类器上，并选择效果最佳的分类器。在不同分类器下的检测精确率、召回率和 F1 值如表 3-6 所示。表 3-6 中的数据表明，在数据集和特征相同的条件下，随机森林在所有指标上均表现出更优的性能，其

中，F1 值相较于逻辑回归提高了 12%，相较于 SVM 提高了 1%。

表 3-9 不同算法检测结果对比

Table 3-9 Comparison of Detection Results Using Different Algorithms

Algorithm	Recall	Precision	F1 score
Logistic Regression	0.86	0.81	0.83
SVM	0.94	0.95	0.94
Random Forest	0.94	0.97	0.95

除了上述检测结果的正确性外，针对本文的目标场景，检测的资源消耗也是重要的指标。针对资源消耗，本文从运行检测框架所需要的内存和检测时长两个维度与文献[51]采用的基于 DNS 流量的检测方法进行了比较。为了验证本文方法的性能，使用了相同数量的实验数据集。结果如表 3-10 所示。从表中数据可见，与文献[51]相比，本文方法在检测时间和内存消耗方面都取得了较好的实验效果。

文献[51]虽然准确率较高，但其检测方法较为复杂，包括关联分析、特征提取，并且基于深度学习和机器学习结合的方法进行检测，因此开销较大。本文在检测方法上进行了优化，只需先聚类然后训练随机森林分类器即可。并且，在特征提取之前，对域名进行了预处理，有效减少了内存开销和检测时间。同时，通过重复实验，在保证实验效果的前提下，去除了冗余特征，保留了影响实验效果的关键特征。

表 3-10 检测性能比较

Table 3-10 Comparison of Detection Performance

DGA 检测方法	检测时间/min	内存开销/MB	准确率
文献[51]	8.76	98	0.96
本文	3.32	2.93	0.93

尽管本文方法的检测准确率并未比已有方法高，但在准确率相近的情况下，检测时间和内存消耗都有显著减少，因此，在针对目标的应用场景下，本文的检测方法具有一定的优势。

3.3 本章小结

在本章中，本文对 DGA 恶意域名检测的方法进行了系统的构建和实现。首先，本文明确了检测的目标效果，并详细阐述了检测方法的总体架构。接着，在检测部分，本文采用了先聚类再分类两种方法来实现对DGA恶意域名的识别。通过聚类算法，本文将相似的域名归为一类，从而揭示出DGA恶意域名家族的内在结构和关系。通过分类算法，本文对聚类后的域名进行进一步的分类，从而实现对不同类型的DGA恶意域名家族的准确识别。在实验部分，说明实验环境信息，并详细介绍了数据集的准备过程，包括数据的具体来源及数量级。对构建的检测方法进行了实验验证，并对实验结果进行了分析。实验结果表明，文本构建的检测方法能够达到预定的目标效果，即实现对DGA恶意域名的准确、高效的识别和分类。

4 DGA 恶意域名检测评估框架

在上一章所述的检测方法基础上，本章设计了评估框架并引入运营评估中心。该中心依据系统在实际应用场景中运营人员所关注的特定指标，构建了一套全面的评估体系，实时更新评估数据，便于运营人员及时掌握检测效果，持续优化检测效果，确保其在实际应用中的有效性和可靠性。并且，为了增强决策支持能力和提升用户体验，完成了运营中心评估结果的动态可视化展示。

4.1 评估框架设计

由于 DGA 恶意域名的发展演变速度迅猛，且网络安全检测系统的专业性较强，因此，在关注系统自身的安全检测能力的同时，还需要安全运营人员对检测模型进行定期评估，以确保检测系统的可用性，包括特性相关性、检测结果准确性等。现有的评估指标大多是对算法的效果说明，缺少反馈机制，且鉴于本系统采用线下部署形式，无法实现云端的实时同步更新，因此，本系统基于评估框架设计运营评估中心，以实现评估部分的功能。检测分析中心与运行评估中心的协同关系如图 4-1 所示。

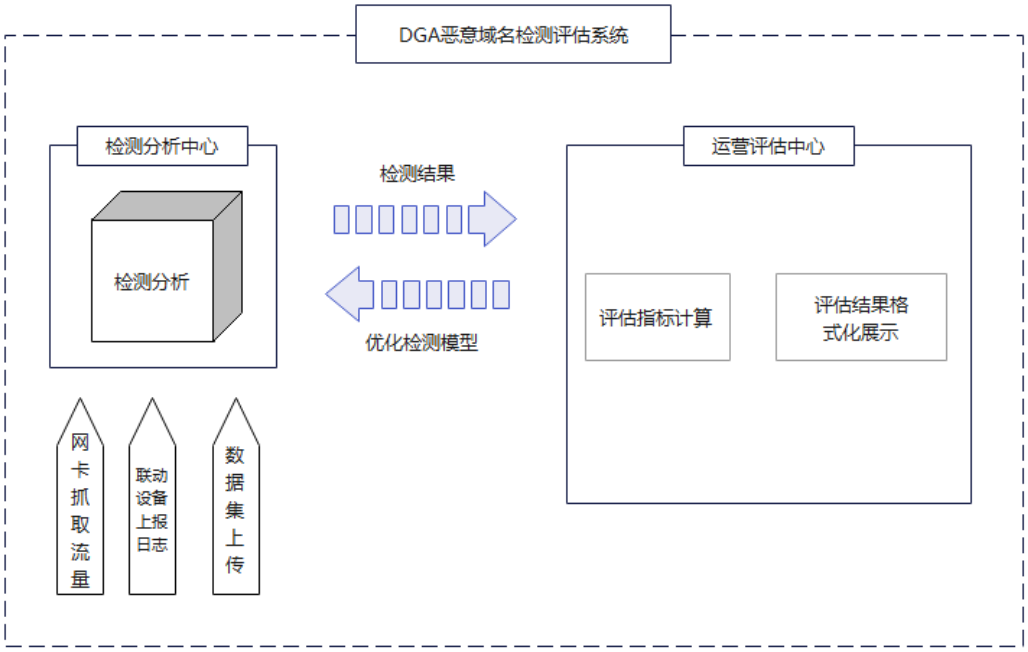

图 4-1 检测与评估协同关系图

Figure 4-1 Detection and Evaluation Collaboration Diagram

在运营评估中心中，可以针对检测方法输出的结果进行相关指标评估，将评估结果反馈给检测模型并可视化的展示给运营人员，达到不断的完善检测方法的目标。基于本系统的目标检测场景，以及结合DGA域名变换频繁的特点，评估框架选取特征相关性、混淆矩阵、PRC、ROC 作为评估指标，设计思路包括以下两点：

（1）当前系统较为关注算法使用的特征是否在频繁变化的 DGA 域名中仍然有效，因此选取特征相关性评估作为评估指标之一，判断特征之间是否存在强烈的多重共线性问题，这项评估表明它们能够独立提供有用的信息。根据评估结果，通过去除不相关或冗余的特征，可以简化模型，减少计算量，加快检测速度，同时提高模型的泛化能力。

（2）鉴于系统的使用场景，在误报率和性能上要求较高，混淆矩阵与 PRC 代表了在低误报率的前提下获得高召回率的重要性。这在要求低误报的场景中尤为重要。ROC 曲线下的面积（AUC）提供了一个单一的数值来总结模型的性能。

4.2 评估框架实现

4.2.1 特征相关性评估

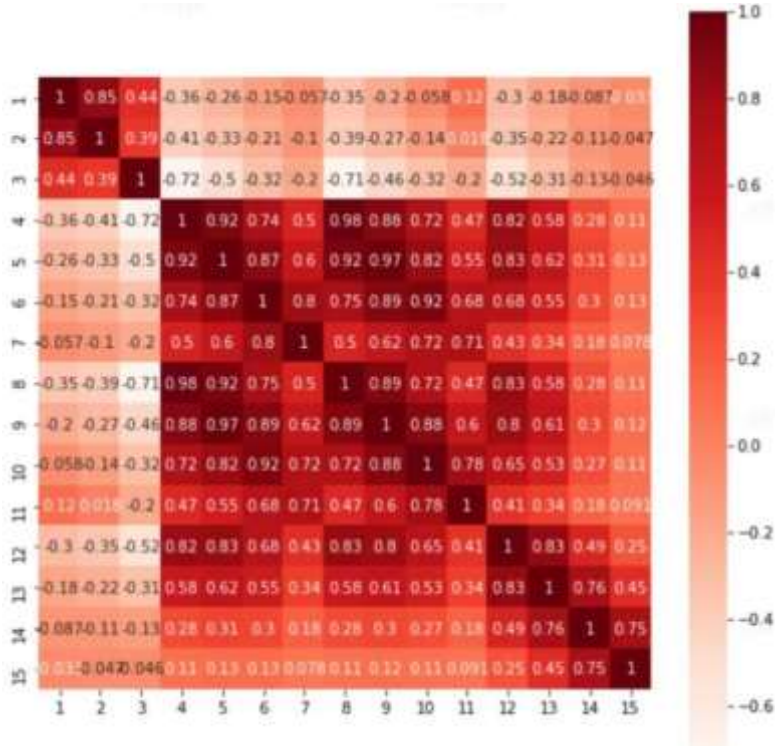

图 4-2 相关性热力图

Figure 4-2 Correlation heat map

为了验证这些特征的有效性，本文进行了相关性分析，以评估它们之间的相

关性程度。通过计算特征之间的皮尔逊相关系数，皮尔逊相关系数的绝对值在0.3 以下被认为是无关或者弱相关，0.3~0.5 属于低相关，0.5~0.9 属于中等程度相关，0.9 以上属于强相关，本文发现这些特征之间大部分不存在强烈的多重共线性问题，表明它们能够独立提供有用的信息，能够代表域名的独特属性。特征的相关性可视化热图如图 4-2 所示。

针对特征重要性的评估方面，通过在训练集上训练一个逻辑回归模型，并计算每个特征的系数大小，进一步确认了这些特征的重要性。逻辑回归模型的系数大小与特征的重要性成正比，实验分析表明，这些特征对分类任务有显著的贡献。同时，考虑了特征的稀有性，确保每个特征在数据集中都有足够的分布，以避免分类任务中的过拟合问题。

4.2.2 检测性能评估指标

本研究所做的 DGA 恶意域名检测，目标是将 DNS 日志流量分为正常域名或由DGA生成的恶意域名，分类结果中仅涉及正确或错误两种情况，因此采用了二分类的准确性评估标准。在二分类中，一个样本被划分为正类和负类。对于异常 DNS 流量的识别，每个样本都具备两个标签：一是其已知的真实类别标签，二是通过训练和测试赋予的预测标签。当这两个标签一致时，表明预测正确；预测正确的比率即为评价分类器优劣的指标。

针对检测结果的的性能评估，采用混淆矩阵实现，混淆矩阵（Confusion Matrix）如下图 4-3 所示，是一种用于评估分类器性能的表格，它展示了实际类别和预测类别之间的关系。

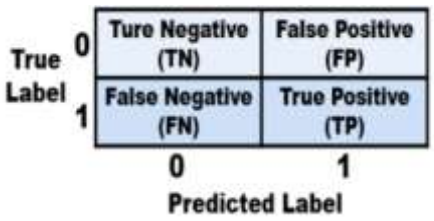

图 4-3 混淆矩阵

Figure 4-3 Confusion Matrix

在二分类问题中，混淆矩阵通常是一个 2x2 的表格，包含四个格子，分别对应四个类别：真正例（True Positive, TP）、假正例（False Positive, FP）、真反例（True Negative, TN）和假反例（False Negative, FN）。针对这四个类别下面给出简单的介绍：

（1）真正例（True Positive, TP）：实际类别为正例，预测也为正例。

（2）假正例（False Positive, FP）：实际类别为反例，预测为正例。

（3）真反例（True Negative, TN）：实际类别为反例，预测也为反例。

（4）假反例（False Negative, FN）：实际类别为正例，预测为反例。

由此，可以引出四个比例，True Positive Rate、False Positive Rate、True Negative Rate、False Negative Rate，这四个指标可以用来评估分类器的性能，并帮助确定其在实际应用中的效果。本文可以根据具体应用的需求来平衡这些比例，以获得最佳的整体性能。混淆矩阵的四个比例计算公式如下：

真正例率（True Positive Rate, TPR）也称为召回率（Recall），是真正例（True Positive, TP）与真正例（True Positive, TP）加上假反例（False Negative, FN）之和的比例。用于表示敏感度或灵敏度，它衡量的是分类器正确识别正例的能力。TPR 越高，表示分类器对正例的识别越准确。即召回率越高，分类器识别真正例的能力越强。计算公式见式（4-1）

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \qquad （式 4-1）$$

其中，TP是真正例，为正类且预测为正类的样本数；FN是假反例，为正类但预测为负类的样本数。

假正例率（False Positive Rate, FPR）即 1 - 特异性（1 - Specificity），它衡量的是分类器错误地将负类预测为正类的频率。FPR 越低，表示分类器对负类的预测越准确，产生假正例的可能性越小。计算公式见式（4-2）：

$$1 - \text{Specificity} = \text{FP}/(\text{FP} + \text{TN}) \qquad （式 4-2）$$

其中，FP是假正例，即为负类但预测为正类的样本数；TN是真反例，即为负类且预测为负类的样本数。

真反例率（True Negative Rate, TNR）用于衡量分类器在预测负类样本时的正确率，即实际为负类且被预测为负类的样本的比例,TNR越高，表示分类器对负例的识别越准确。其公式如下：

$$\text{TNR} = \text{TN}/(\text{TN} + \text{FP}) \qquad （式 4-3）$$

其中，TN是真反例，即实际为负类且预测为负类的样本数；FP是假正例，即实际为负类但预测为正类的样本数。

假反例率（False Negative Rate, FNR）用于衡量分类器在预测正类样本时的错误率，即实际为正类但被预测为负类的样本的比例。FNR 越低，表示分类器对正类的预测越准确。见式（4-4）

$$\mathrm{FNR} = \mathrm{FN}/(\mathrm{TP} + \mathrm{FN}) \qquad （式 4-4）$$

其中，FN是假反例，它表示为正类但预测为负类的样本数；TP是真正例，它表示为正类且预测为正类的样本数。

上述四个指标在混淆矩阵中至关重要，除此之外，在 DGA 检测的评估框架中，还涉及到三个关键指标，分别是精确率（Precision）、准确率（Accuracy）、F1 分数（F1 Score）。

精确率衡量的是在所有预测为正类的样本中，真正例的比例，精确率反映了分类器在正类样本上的准确性，即预测为正类的样本中，真正例的比例。准确率衡量的是分类器正确分类所有样本的比例，反映了分类器在整个数据集上的性能，即所有样本中正确分类的比例。F1 分数是精确率和召回率的调和平均数，它综合考虑了分类器的精确率和召回率，它是一个平衡指标，它既考虑了分类器在正类样本上的性能，也考虑了分类器在负类样本上的性能。涉及的公式如下：

$$\mathrm{Precision} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP}) \qquad （式 4-5）$$

$$\mathrm{Accuracy} = (\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}) \qquad （式 4-6）$$

$$\mathrm{F1} = 2 \times (\mathrm{Precision} \times \mathrm{Recall})/(\mathrm{Precision} + \mathrm{Recall}) \qquad （式 4-7）$$

精确率、准确率和 F1 分数都是评估分类器性能的重要指标。精确率关注的是预测为正类的样本，准确率关注的是整个数据集，而 F1 分数则综合考虑了正类和负类的性能。

考虑到 DNS 日志内容中，合法域名与恶意域名分布不均的情况，在评估指标中，引入 AUROC（Area Under the Receiver Operating Characteristic）和 AUPRC（Area Under the Precision-Recall Curve），用于衡量分类器性能的指标。AUPRC 是 Precision-Recall 曲线下面积的度量，它表示分类器在所有可能的召回率水平下的平均精确率。曲线下面积 AUC(Area Under Curve)越大,说明模型性能越好。

4.3 评估框架验证

4.3.1 实验环境

本实验所使用的硬件配置参数及软件的版本信息如表 4-1 所示：

表 4-1 实验环境说明

Table 4-1 Experimental Environment Description

环境配置项	详情
CPU	Intel(R) Core(TM) i7-9700K@ 3.60GHz
内存	16GB DDR4
操作系统	Linux
开发语言	Python 3.8

4.3.2 数据集

本文在此评估框架下，选取了四种样本集，作为检测模型的输入，样本的方案分别采用训练用的黑白样本，与未使用过的黑白样本交叉组合生产。白样本分别来源于 Alexa、Tranco，黑样本来源于 360 网络安全实验室以及恶意家族算法生成，具体样本组合如表 4-2：

表 4-2 评估样本数据集

Table 4-2 Evaluation Sample Dataset

样本方案	样本集组成
样本 1	100 万训练白样本、265474 训练用黑样本
样本 2	100 万训练白样本、1354868 个黑样本
样本 3	100 万训练白样本、3542456 个黑样本
样本 4	3548745 个白样本、3542456 个黑样本

4.3.3 可视化展示结果

依次使用四个样本方案进行检测，并使用上文中的评估指标，计算检测模型中各项指标的表现，其混合矩阵如图 4-4 所示。

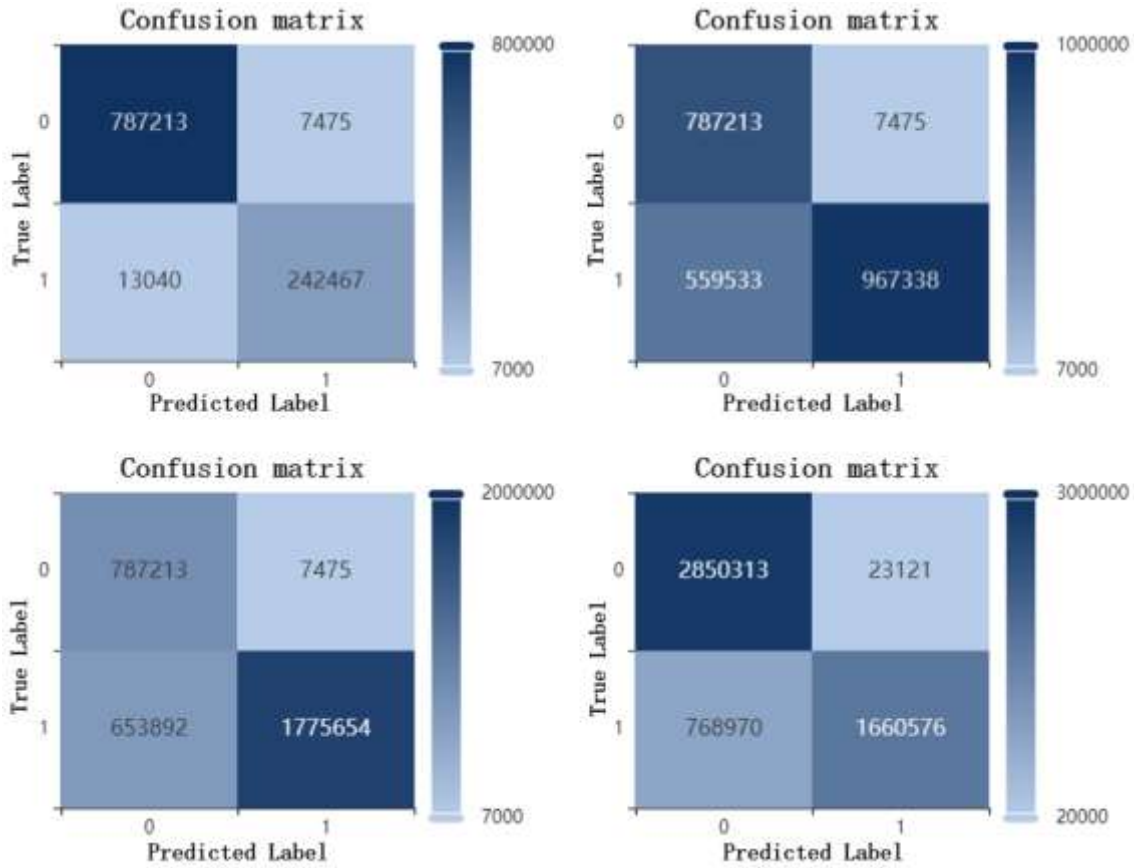

图 4-4 样本 1~4 混淆矩阵可视化结果

Figure 4-4 Sample 1~4 confusion matrix visualization results

PRC 和 POC 曲线如图 4-5~4-8 所示：

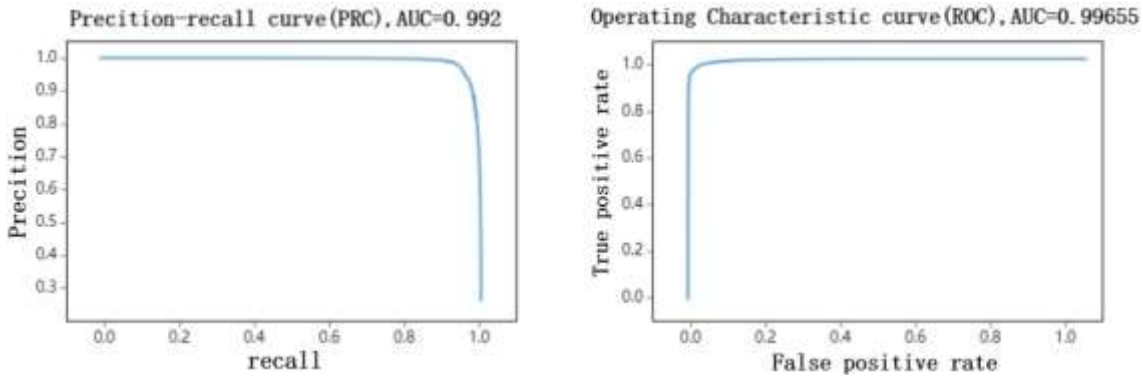

图 4-5 样本 1 PRC 和 POC 曲线

Figure 4-5 Curve of PRC and POC for Sample 1

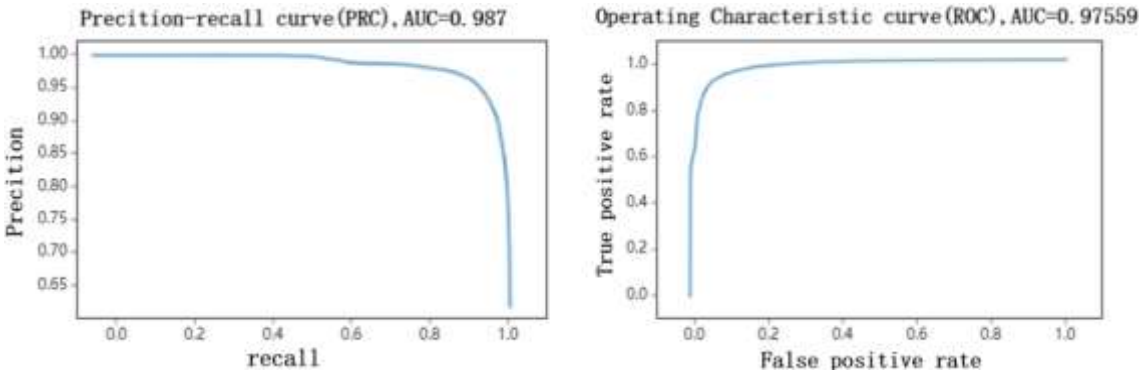

图 4-6 样本 2 PRC 和 POC 曲线

Figure 4-6 Curve of PRC and POC for Sample 2

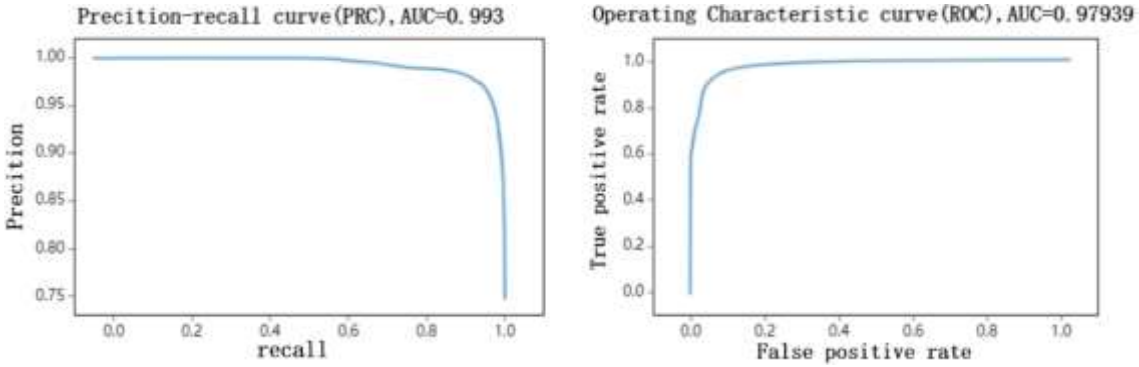

图 4-7 样本 3 PRC 和 POC 曲线

Figure 4-7 Curve of PRC and POC for Sample 3

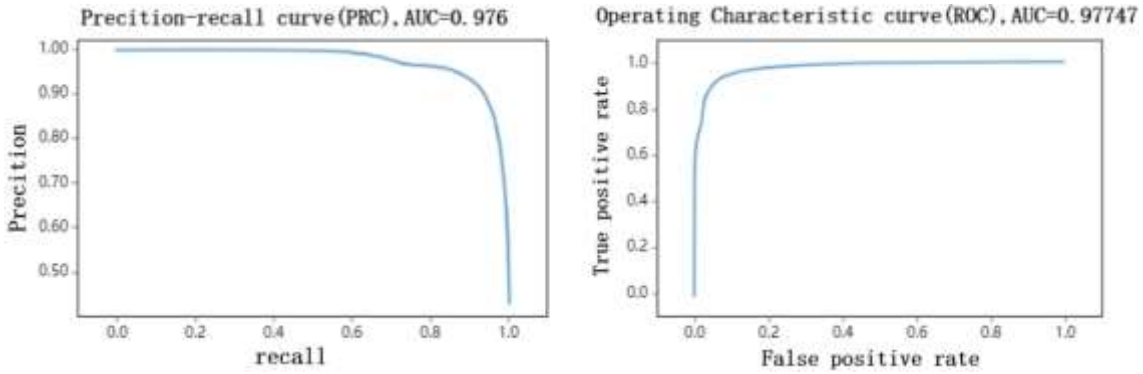

图 4-8 样本 4 PRC 和 POC 曲线

Figure 4-8 Curve of PRC and POC for Sample 4

AUPRC 和 AUROC 在四类样本集的评估测试中，均在 0.97 以上，反映了模型本身具有良好的分类性能。较高的 AUPRC 反映了模型在不平衡的测试样本

集中依然有较好的分类效果。较高的 AUROC 反映了模型在阈值调整时分类性能的潜力较高，模型分类能力较好，在不同数据集上的评估结果如表 4-2 所示。

表 4-2 评估样本数据集

Table 4-2 Evaluation Sample Dataset

指标	样本 1	样本 2	样本 3	样本 4
AUPRC	0.992	0.987	0.993	0.976
AUROC	0.997	0.976	0.979	0.977
Accuracy	0.980	0.872	0.901	0.963
Precision	0.970	0.992	0.996	0.986
Recall	0.949	0.638	0.731	0.683
F1 score	0.959	0.777	0.843	0.807

4.4 本章小结

在本章节中，目的在于构建一个全面的 DGA 恶意域名检测结果评估框架，对检测结果的质量及性能做出评价，以提高检测方法的性能和可靠性。首先，基于皮尔逊相关系数对特征相关性进行了评估，并通过热图展示了特征之间的独立性，从而为检测方法优化提供了重要参考。接着，制定了一系列评估指标，这些指标旨在量化检测方法的性能，包括准确率、召回率、F1 分数等。通过对这些指标的评估，本文能够准确地了解检测方法的优劣，并为改进提供方向。最后，利用可视化工具，将评估结果直观地呈现出来，以便研究人员和从业者能够更好地理解检测方法的性能，并对其进行持续优化。

5 DGA 检测与评估系统的设计与实现

随着僵尸网络技术的持续演进，其对社会的潜在威胁日益凸显。本章的目标是构建一个基于本地服务器的恶意 DGA 域名检测与评估系统，旨在整合前文提出的 DGA 域名检测方法与评估框架，设计并实现一个专门针对“On-Premises”模式的 DGA 域名检测与评估系统。该系统能够实时监测并分析监控组网中边界设备上报的 DNS 日志流量，检测结果可作为后续防护动作的依据，有效拦截针对 DGA 域名的解析尝试，阻断僵尸网络的命令控制信道。此外，系统基于前文的评估框架，实现了运营评估中心，持续优化检测方法，确保其在实际应用中的有效性和可靠性。最终，系统将域名检测与评估的详细信息通过前端的可视化界面展示，使得安全管理人员能够持续关注网络中的 DGA 恶意域名，并及时对异常主机采取相应的防护措施。

在具体实现上，本系统采用了 Apache Spark 这一高性能的分布式数据处理框架，有效实现了对大规模数据的并行处理。通过将请求及数据分布于集群中的多个节点，Spark 的内存计算模型显著减少了磁盘 I/O 操作，极大地提升了数据处理的速度和效率。此外，本研究针对政企、医疗等对数据隐私要求较高的应用场景，因此，结合 Hadoop 的分布式存储系统，成功实现了本地部署的 DGA 检测与评估系统。该系统能够在不涉及数据上云的情况下，保障数据的安全性和隐私性，满足特定行业对数据处理的需求。

5.1 系统需求分析

在构建 DGA 检测系统之前，首先需要先明确系统的应用场景以及需要解决的问题，并对这些问题的紧迫性进行排序。这些问题可以根据其重要性分为三个等级：关键、重要和次要。随后，针对每个等级的问题，设计相应的系统模块，从而将系统划分为多个功能性的组件。在初始阶段，选择最核心的几个模块来开发最小可行性产品（MVP），而其他模块则可以根据业务发展的需求逐步完善和升级。

5.1.1 应用场景分析

DGA 域名检测与评估系统可以作为一个独立的安全检测模块部署，在通过对内网流量的检测与评估后，给出检测结果，起到预先防御的功能，本系统适用于

对数据隐私性要求较高的场景，针对政务、医疗、金融等对数据私密性要求较高的场景，本系统通过本地部署方式，能够满足这些场景对数据安全和隐私的需求。系统部署在他们的内网环境中，可以完成恶意域名的检测和防护，同时具备高性能、快速响应和良好的鲁棒性特点。

政务机构处理大量敏感数据，如公民个人信息、政府政策文件等，因此对数据安全和隐私保护有极高的要求。政务系统需要一个本地部署的恶意域名检测系统，以确保政府网站和内部网络的安全，防止恶意域名攻击和信息泄露。同时，系统的高性能和快速响应能够满足政务系统对实时性的要求。

针对医疗系统中，国内大部分医院禁止医疗数据上云，医疗系统存储着患者的个人信息、病历记录等敏感数据，一旦数据泄露，可能对患者隐私造成严重威胁。但是医院的数据体系是非常庞大的，因此需要一种可以具备大数据存储能力的安全防御系统，本系统的本地部署方式能够确保医疗数据不离开内网，减少数据泄露的风险。同时，系统的快速响应能力能够及时发现并阻止恶意域名攻击，保护医疗系统的网络安全。

5.1.2 部署需求分析

系统部署需求主要包括本地部署、硬件要求、软件要求、网络配置、系统集成五个方面：

（1）本地部署：系统需要部署在客户本地网络环境中，以满足对数据私密性和安全性的要求。部署方式包括在客户内部服务器或专用硬件设备上安装系统。且采用一键式安装，提高工程效率，后续可以通过打补丁方式对系统进行修复或优化。

（2）硬件要求：系统部署所需的硬件设备应具备足够的计算能力和存储空间，以支持大规模数据处理和存储。硬件设备应满足以下要求：CPU：至少四核以上，建议使用多核处理器以提高系统处理速度； 内存：至少 8GB 以上，建议使用 16GB 或更高以支持大数据处理；存储：至少 1TB 以上，建议使用分布式存储系统以支持大规模数据存储；网络：至少千兆以太网接口，建议使用冗余网络接口以提高系统可靠性。

（3）软件要求：系统部署所需的软件环境应满足在 Linux 操作系统上进行系统部署和搭建，开发环境支持 Python、Java 等编程语言，用于实现系统功能和算法。框架和库支持 Spark、Hadoop 等分布式计算框架，以及 PyTorch、TensorFlow 等机器学习库，用于实现大规模数据处理和分析。

（4） 网络配置：系统部署时，需要进行网络配置以确保系统之间的通信和

数据传输。网络配置应满足以下要求：内网隔离，确保系统部署在内网环境中，与其他网络环境隔离，以提高系统安全性；保持与联动设备的连通性，以便可以获取到内网中防火墙上报的日志流量，针对有端口阻断的网络环境，需要对接收端口配置安全组，防止上送数据时被异常阻断。

（5）系统集成：系统需要与客户现有的 IT 基础设施和业务系统进行集成，以实现数据共享和业务协同。集成需求应满足数据接口标准化，提供标准的数据接口和 API，以便与客户的其他系统进行数据交换和集成。业务协同方面，应支持与客户的其他业务系统进行协同工作，如报警系统、日志系统等，以实现业务流程的自动化和优化。

通过满足上述部署需求，本系统将能够高效、稳定地部署在客户的本地网络环境中，为客户提供高可靠性的恶意域名检测和防护服务。

5.1.3 功能需求分析

基于上述的应用场景，本系统的功能需求至少包含五个部分，即数据采集模块、数据处理模块、数据检测与评估、数据存储、可视化展示，具体需求分析如下：

（1）数据采集模块：此模块负责收集 DNS 域名数据，并根据不同的应用场景，支持来源于防火墙等边界设备或者终端设备的数据上报，也支持来源于前端页面客户的域名查询。在特定的测试环境中，用户可能需要查看开源于不同的主机的域名检测结果，因此该模块应具备根据参数进行域名检测记录筛选的能力。

（2）数据处理模块：该模块负责对采集到的数据进行必要的处理，为了提高数据采集的准确性和效率，需要对采集到的数据进行质量检测和清洗。由于域名的来源有多种渠道，所以需要数据处理模块满足用户不同数据来源的数据收集，对数据格式进行归一化，满足对异常数据的分析和处理需求。其功能包括将数据中没有主机域名的数据去除掉。

（3）DGA 检测与评估模块：鉴于本研究聚焦于恶意域名攻击，恶意检测模块的设计旨在高效、可靠的将域名进行区分，包括恶意 DGA 域名和合法域名两种，检测模块需要具备处理大规模数据集的能力，能够对归一化后的数据进行高效的分析和处理。检测模块应能够准确识别和分类 DGA（Domain Generation Algorithm）恶意域名。模块中集成的检测模型具有较高的准确率和较低的误报率。由于数据的输入方式有两种，因此检测模块应支持实时数据处理，以便快速响应新的恶意域名活动。同时，也应支持批量处理模式，以处理历史数据和大规模的数据集。在评估部分应基于评估框架中的指标参数，用于对检测模型的效果进行

量化评估。评估指标应包括但不限于准确率、召回率、F1 分数等，并可以存储提供评估结果，以便于审查和回溯。

（4）数据存储：存储模块需能够处理和存储大规模的检测数据，包括但不限于域名信息、网络原始流量数据、检测结果等。随着系统运行时间的增长，数据量会持续增加，因此存储模块必须具备可扩展性，能够随着数据量的增长而动态扩展存储容量。并且由于目标系统对数据的持久化要求较高，存储模块必须保证数据的高度可靠性，防止数据丢失或损坏。通过 Hadoop 的分布式文件系统（HDFS）[52]实现数据的冗余存储，确保即便在单个或多个存储节点发生故障的情况下，数据依然可用。考虑到未来的扩展性，存储模块应具有良好的兼容性，能够与现有的 IT 基础设施和技术栈集成。应具备良好的可扩展性，能够支持未来可能引入的新技术和新功能。

（5）可视化：针对检测的 DNS 流量，监控模块需要能够将检测结果通过可视化的方式呈现给用户。这包括图表、热图等多种形式，以便用户能够直观地理解检测数据。除了可视化展示外，监控模块还应提供详细的检测数据展示功能。这包括展示恶意域名的详细信息，如域名名称、检测结果，近期趋势等。

通过上述模块的设计和实现，本系统旨在为用户提供一个全面、高效的 DGA 恶意域名检测评估平台，以增强网络安全防护能力。同时，系统的可扩展性和可定制性使得用户可以根据自己的需求进行功能扩展和优化，以适应不断变化的网络安全威胁。

5.1.4 非功能需求分析

评价一个系统的优劣，不能仅基于其功能是否齐全或是否达到使用目标，这样的评价既不全面也不够严格。在系统设计过程中，还必须综合考虑系统的稳定性、响应速度、安全性和可扩展性等非功能性需求。因此，本系统的设计考虑了以下非功能性需求：

（1）系统稳定性：作为一种安全防御软件，本系统需能够长时间连续运行，以实现对网络设备和设施的实时保护。考虑到系统可能部署在数据中心、学校机房等无人监管的环境中，系统的稳定性至关重要。系统崩溃可能导致网络保护能力的丧失，造成严重后果。

（2）系统可靠性：在实际运行环境中，DGA 检测系统可能面临多种潜在的故障和异常情况，包括但不限于异常下电、硬件故障、软件错误、网络中断等。为了确保系统在遇到这些情况时能够保持稳定运行，满足业务需求，并达到预定的最小 SLA，并在故障解除后，业务恢复正常。

（3）系统安全性：系统安全性包括抵御针对系统本身的攻击行为以及保护数据安全。系统应具备检测异常的同时，也要能够抵御对自身的攻击。对外暴露的接口，要防止 XSS 等注入类攻击。此外，还需要对涉及到用户隐私信息的存储，进行匿名、假名处理。

（4）系统可扩展性：系统的可扩展性是设计时必须考虑的因素之一。随着系统漏洞的发现或新功能的增加，系统升级是不可避免的。良好的可扩展性可以减少开发时的重构和维护工作，提高开发效率。

5.2 系统架构设计

本文设计的恶意域名检测系统的架构采用了 MVC（Model-View-Controller）设计模式，该系统由数据层、控制层和视图层三个层次组成，如图 5-1 所示。在数据层中，负责接收流量和日志、存储样本数据、检测结果等。控制层负责系统核心业务逻辑的具体实现，基于 Spark 实现大规模的数据处理,负责接收用户请求并调用检测评估模型完成处理，然后将结果传递至视图层。视图层是用户交互的界面，承担着展示检测检测结果、评估图表、系统配置等功能。MVC 架构通过将前端展示和后端逻辑分离，以及将功能模块进行层次化封装，增强了系统的灵活性，便于各模块的迭代和更新。在系统底层，使用 Spark 实现大数据下的 DGA 检测，Hadoop 的 HDFS 被用来存储原始日志、流量及评估结果等。

系统运作流程如下：首先，由数据层的数据采集模块负责从多个防火墙或终端设备主动搜集相关数据，如域名信息等。数据采集模块也可以从网卡抓取流量并转换成日志格式。由于数据的来源不同，因此这些数据需要被送入到数据预处理模块进行归一化处理 ，此模块主要负责对原始 DNS 域名数据进行解析、清洗和过滤，并对日志格式进行归一化，归一化的数据采用统一的大字段形式存储，处理后的数据传输到检测模块进行 DGA 恶意域名检测，检测模块基于 Spark 实现大规模的数据处理。Spark 提供了快速的数据处理能力，能够有效地处理和分析大规模的数据集。在这一层采用前文构建的 DGA 恶意域名检测模型及评估框架，通过该模型对数据进行分析和分类，判断域名是否存在恶意行为。评估模块用于评价系统的效果和价值，不仅包括精确率、F1 值等用于评价算法模型的指标，还包括当前特征的有效性。在上述过程中产生的原始流量、原始日志、检测结果将由存储模块进行存储，存储模块基于 Hadoop 实现分布式存储，达到大数据的持久化目标，Hadoop 的分布式文件系统（HDFS）具有高可靠性和可扩展性，能够存储大规模的数据集，并保证数据的可靠性和可服务性。增加系统鲁棒性及安全性。另外，考虑到系统日志审计的安全性，系统的相关操作日志也将存储在此模块，

以便后续的分析和审计。经过检测分析的结果通过前端展示页面可视化的呈现给用户，包括检测评估结果的图表和详细的检测数据，系统也可以提供自定义时间及域名的模糊检索，使用户能够直观地了解当前域名的情况。整个系统的设计旨在提高恶意域名检测的精确性和效率，并为用户提供一个安全性好、鲁棒性高的安全检测系统，为网络安全的维护提供有力支持。

图 5-1 恶意域名监测系统框架

Figure 5-1 Malicious domain name detection system framework

5.3 模块设计与实现

5.3.1 数据采集模块

该模块的主要职责是从多个数据源收集网络流量数据，以实现对潜在恶意域名活动的全面监控。本模块的功能主要涉及两个方面。首先，数据采集模块提供了灵活的前端配置功能，允许用户在界面中配置联动设备的信息。这包括设备的型号、IP 地址、端口号等关键参数。一旦用户完成配置，采集模块便能够接收并解析对应设备上送的日志数据，从而实现对日志的自动化处理。这一功能不仅简化了用户的操作流程，也提高了系统对日志数据的处理效率。其次，数据采集模块还负责从网卡接收流量，使用 tcpdump 从网卡捕获流量,tcpdump 是 Linux 系统中一个常用的抓包命令。tcpdump 能够截取网络流量数据并保存，同时支持主机、端口等参数的过滤。在采集 DNS 流量时，使用特定的命令配置 tcpdump 以实现目标。具体命令为：tcpdump -i eth0 -nt -X port domain -w dns-dump.pcap -v。其中，使用的参数及其含义如表 5-1 所示。

表 5-1 tcpdump 命令参数说明

Table 5-1 tcpdump Command Parameters Explanation

-i interface	指定要捕获的网络接口
--numeric	以数字形式显示主机和端口地址，不解析主机名
-w file	将捕获的数据包保存到文件中
-s len	设置捕获数据包的长度
-X	详细显示数据包的内容

捕获过滤后的数据由 Kafka 消息系统进行队列控制,传输到数据预处理及检测评估模块进行处理，接收到的数据是 pcap 包格式的流量数据。pcap 包是网络数据包的原始形式，通常包含大量冗余信息，直接用于检测和处理较为复杂。因此，采集模块具备将 pcap 包转换为日志数据的能力，这一转换过程有助于简化数据处理流程，并为后续的数据预处理和模型构建提供支持。

5.3.2 数据处理模块

该模块的核心功能是对数据进行预处理，以提升其质量并确保其适合后续的数据分析和模型构建。尽管数据采集模块已经将原始流量数据转换为日志格式，但数据本身仍需经过一系列的处理步骤，以实现数据的优化和标准化。数据清洗是数据预处理的重要环节，它旨在识别并纠正数据集中的错误和不一致性。这可能包括处理缺失值、去除噪声数据、转换数据格式等。数据去重则用于识别并移除数据集中的重复记录，以确保数据的唯一性和准确性。数据过滤则用于筛选出与恶意域名检测相关的关键信息，如源目的 IP 地址、域名信息等。

通过这些预处理步骤，数据处理模块能够从日志中提取出有价值的信息，并将其转换为数据检测模块可以处理的数据格式。例如，域名信息需要经过特定的处理，如去除顶级域名，以便于数据检测模块能够准确地识别和分析恶意域名。综上所述，数据处理模块在恶意域名检测系统中发挥着至关重要的作用。通过数据清洗、数据去重和数据过滤等预处理步骤，数据处理模块能够优化数据质量，确保数据的一致性和准确性，并为数据检测模块提供高质量的数据输入。这一过程对于构建准确和高效的恶意域名检测系统至关重要。

5.3.3 检测与评估模块

在本系统中，检测模块依托 Spark 框架，本系统采用三节点部署形式，实现了对大规模数据集的高效处理。Spark的并行计算特性使得数据处理变得快速而高效，能够有效地分析和处理大规模的数据集。在这一层，系统采用了前文构建的DGA 恶意域名检测方法及评估框架，通过该模型对数据进行深入的分析和分类，以判断域名是否存在恶意行为。

在运营中心实现评估模块，中心依据系统在实际应用场景中运营人员所关注的特定指标，构建了一套全面的评估体系。系统支持从前端页面接收运营人员上传的数据集，并根据评估框架可视化展示评估指标，实时更新评估数据，便于运营人员及时掌握检测效果，持续优化检测效果，确保其在实际应用中的有效性和可靠性。

5.3.4 存储模块

在现实网络环境中，DNS 数据的规模庞大，每日产生的日志数据量约为836.17G，DNS 报文数据量高达 4.91 亿，这对存储能力提出了较高的要求，需要确保数据的可靠性和可访问性。为了应对这一挑战，本系统的存储模块采用了Hadoop 分布式存储架构，实现了大数据的持久化目标。Hadoop 的分布式文件系统（HDFS）具备高可靠性和可扩展性，能够存储大规模的数据集。本系统采用了三节点部署方式，以增强系统的鲁棒性和安全性。

如图 5-2 所示，存储模块主要包括对原始流量、原始日志和检测结果等内容的存储。原始流量和原始日志主要保存了采集器采集到的不同数据源的数据。其中，原始流量针对网卡抓包数据，而原始日志则是接收的不同联动设备上报的日志信息，如防火墙、终端防护设备等。这些流量信息在 HDFS 中以分布式形式存储，以保证数据的可靠性和可访问性。

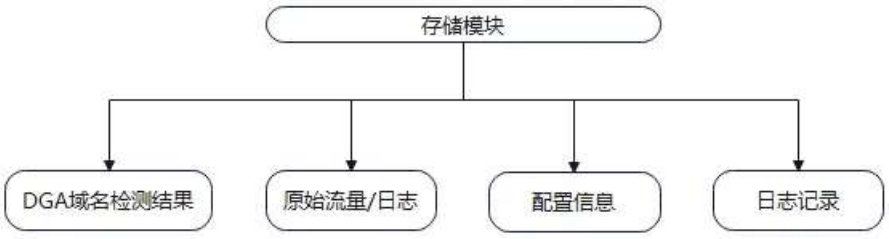

图 5-2 存储模块

Figure 5-2 Storage Module

除了上述数据信息，考虑到系统日志审计的安全性，系统的相关操作日志也

将存储在此模块，以便后续的分析和审计。此外，用户的登录信息以及前端页面下发的配置信息也会进行存储，以保证系统的正常运行和数据的安全性。

5.3.5 展示模块

该模块基于 VUE 实现，功能设计主要分为四个部分：检测结果列表展示、近期趋势分析、联动设备配置以及恶意域名情报查看。这些功能共同构成了系统的用户界面，旨在提供直观、实时的信息展示，同时增强用户对系统运作的理解和操控能力。

首先，检测结果列表展示了当前系统中正在进行的域名检测活动。此功能实时地在前端页面上展示检测列表，对于检测结果显示为恶意的域名，系统将进行实时标注。这些标注内容包含域名信息、源/目的 IP 地址、访问次数、时间戳等关键信息，以便用户迅速识别并采取相应措施。同时，对于检测结果为合法的域名，系统将其标记为合法，为用户后续可能的白名单添加等操作提供参考。

其次，近期趋势查看功能允许用户根据自定义时间范围，查看该段时间内系统接收到的 DGA 域名及其统计信息。通过数据可视化技术，系统将以图表的形式展示整体接收数据的趋势，帮助用户把握安全态势，预测潜在威胁。

再次，联动设备配置页面使用户能够配置需要联动的设备。配置完成后，采集模块将能够接收并解析对应设备上送的日志数据，从而实现多设备之间的协同工作，增强系统的整体检测能力。

最后，恶意域名情报查看功能旨在收录当前已知的各类 DGA 及其相应的域名详细信息。这一功能作为科普模块的一部分，便于用户了解 DGA 的种类、特征和相关知识，从而提高用户的安全意识和应对能力。

5.4 系统测试方法

5.4.1 测试方法

(1) 黑盒测试（Black Box Testing）：

属于软件测试技术，其核心思想是将软件视为一个不可见的“黑盒”，这个测试方法仅关注软件的输入输出行为，而不关心其内部实现细节。在黑盒测试中，依据软件需求规格说明书（SRS）设计测试用例，这些用例包含了输入数据和预期输出结果。通过执行这些测试用例，可以验证软件的功能是否符合需求规格，并发现功能相关的缺陷。

黑盒测试的优点在于它能够从用户的角度出发，确保软件的功能满足用户需求。同时，黑盒测试也能够发现一些与功能相关的缺陷，从而提高软件的质量。然而，由于不关注软件内部实现，黑盒测试可能无法发现一些与内部逻辑相关的缺陷。因此，在实际测试过程中，黑盒测试通常需要与其他测试方法相结合，以全面验证软件的质量。

在黑盒测试中，常用的测试策略包括等价类划分、边界值分析、错误推测、因果图、判定表等。这些策略可以帮助测试人员设计出更加全面的测试用例，提高测试覆盖率。例如，在检测系统中，需要进行页面、用户交互接口功能正确性验证。同时，为了提高测试效率，黑盒测试也可以采用自动化测试工具，如Jemeter、Postman，来实现对测试用例的自动化执行。

(2) 白盒测试（White Box Testing）：

又称为结构测试或透明盒测试，是一种软件测试方法，其核心思想是在对软件的内部结构和实现细节有深入的了解的前提下，基于这些知识来设计测试用例，主要会关注软件的内部逻辑、数据流和控制流，以确保代码的每个部分都得到了充分的测试。同时也包含一些编码规范，如代码是否按照设计规范编写。因此本系统的白盒测试是在黑盒测试之前进行的，这样可以尽量在项目早期发现，降低问题的修改成本，提高开发效率。

(3) 安全测试（Performance Testing）：

在DGA恶意域名检测与评估系统中，主要关注用户的个人信息安全，相关信息匿名存储。并且在系统上的操作均记录日志，且记录清楚无歧义，便于在需要澄清或审计时提供凭据。

(4) 可靠性及稳定性测试：

主要关注系统在产生故障时的响应能力，目标是确保DGA恶意域名检测系统能够在预期的工作环境中稳定运行，避免出现故障和错误。通过测试，可以评估DGA 恶意域名检测系统的可靠性和稳定性，并找出潜在的问题和缺陷。测试方法包括压力测试、负载测试、故障注入测试等。这些方法可以模拟不同的工作负载和条件，结合本系统的应用场景。

在可靠性测试中，重点测试了系统对于节点掉电重启、网络延迟、CPU 过载等异常情况下的表现，以及故障撤销后，系统恢复 SLA 的时长。除了异常场景的测试，长稳测试也至关重要，本系统需要长时间服务无间断的运行在服务器上，一旦产生业务中断，可能会造成用户暴露在无安全防护的网络环境中，造成损失。长稳测试共持续 14 天，使用脚本采集环境上的资源占用及服务进程重启等情况。并在长稳期间定时执行高频操作，确保系统可以正常提供服务。测试方法的流程图如下图 5-3 所示。

图 5-3 测试流程

Figure 5-3 Test Process

5.4.2 测试设计及执行

在本章第一节中，详细介绍了系统的需求分析，本节根据这些需求设计了相应的测试用例。这些测试用例覆盖了系统的各个子模块，以及整体系统的性能和功能。在本节中，将基于这些测试用例，对系统进行全面的测试，并对测试结果进行分析。

本系统的环境搭建涉及主要涉及到防火墙和部署了检测系统的 Linux 服务器。防火墙作为系统的第一道防线，能够接收网络流量并生成 DNS 日志，这些日志将被上送到检测服务器中进行 DGA 恶意域名检测。

在服务器部署方面，采用了三节点大数据集群的方式，这不仅提高了系统的处理能力和数据处理速度，还增强了系统的可靠性和容错能力。每个节点都配备了高性能的硬件和足够的存储空间，以满足系统运行的需求。通过 Raid1 技术实现物理备份，能够在发生硬件故障时快速恢复数据，确保系统不会因为单点故障而受到影响。具体测试环境如表 5-2 所示，详细列出了服务器的硬件配置、存储配置以及网络配置等信息。经过测试，这些配置可以确保系统达到预期的性能和可靠性标准。

表 5-2 测试环境配置信息

Table 5-2 Test Environment Configuration Information

设备	硬件配置项	详情
Node1/Node2/Node3	CPU	2×12 核处理器
	内存	256G
	硬盘	2T 数据盘
		600G 系统盘
防火墙	CPU	2×12 核处理器
	内存	256G
	硬盘	2T 数据盘
		600G 系统盘

根据深入的需求分析，明确界定了本系统的核心功能，涵盖了域名检测、检测结果评估、联动设备配置、日志记录、近期趋势分析以及恶意域名情报查询等多个关键环节。为了确保这些功能的有效性和稳定性，本文设计了相应的测试用例，并对每一个用例进行了严格的测试。通过这些测试，本文对系统的各个组成部分进行了全面的验证，结果表明这些核心功能均达到了预期目标，确保了系统的整体性能和用户体验，并且，其能够满足用户需求，并在实际应用中展现出高效、稳定和可靠的特点。

表 5-3 域名检测用例

Table 5-3: Domain Name Detection Use Cases

测试对象	域名检测
测试目的	测试该模块是否可以正常检测域名
预置条件	1. 在目标服务器上完成系统搭建 2. 系统上服务正常 3. 具备可以检测的 Pcap 包数据
操作步骤	在系统的后台节点上执行： curl -H "Content-Type: application/json" -X POST -d '{"do-main":"vojyqem.com"}' http://127.0.0.1:12120/detect
预期结果	{“domain”:”vojyqem.com”,”binary_class”:”0.94512”}
实际结果	测试结果一致

（1）域名检测测试：作为系统的一个核心模块，域名检测部分采用了已经训练好的DGA域名检测评估模型。该模型的结构在第三章节中进行了详细描述。在本部分，本文设计了针对性的测试用例，验证DGA域名检测能力。测试用例如表5-3 所示，根据表 5-3 的测试结果，域名检测模块成功实现了对 DNS 请求中的域名字符串进行DGA恶意域名检测的功能需求。该模块可以基于数据采集模块获取的数据，通过模型的检测，提供域名的二分类的推理值，从而有效识别潜在的恶意域名。测试用例如表 5-3 所示。

（2）检测结果评估测试：基于第四章中构建的评估框架实现，该框架为模型性能提供了全面的评价标准。评估指标包括但不限于精确率（Precision）、F1值等。系统支持针对测试集完成模型评估，通过特征相关性热图、混淆矩阵、精确率和 F1 值等指标，可以直观地反应了模型的检测效果。此外，系统还提供了可视化展示功能，使得测试结果更加直观易懂，帮助运营人员持续优化检测模型。通过可视化展示，用户也可以直观地了解模型的检测效果，从而更好地评估模型的性能。测试用例如表 5-4 所示。

表 5-4 检测结果评估用例

Table 5-4 Detection Result Evaluation Use Cases

测试对象	检测结果评估
测试目的	测试该模块是否可以基于上传到数据集，输出特征相关性热图、混淆矩阵、PRC 及 ROC 曲线图。
预置条件	1. 系统环境已设置完毕，包括所有必要的软件和硬件配置。 2. 第四章中构建的评估框架已集成到系统中。 3. 测试数据集已准备就绪，包括已知恶意和正常的域名样本。
操作步骤	1. 启动系统，并确保系统处于稳定运行状态。 2. 导入测试数据集到系统中，确保数据集的完整性和准确性。 3. 执行第四章中构建的评估框架，对模型进行评估。 4. 记录评估过程中的各项指标，包括精确率（Precision）和 F1 值等。 5. 对比评估结果与预期结果，确保评估框架的准确性。
预期结果	1. 系统能够成功导入测试数据集，并进行模型评估。 2. 评估框架能够准确地计算出精确率（Precision）和 F1 值等指标。 3. 评估结果与预期结果一致，验证评估框架的有效性。
实际结果	测试结果一致，评估图像如图 5-4 所示：

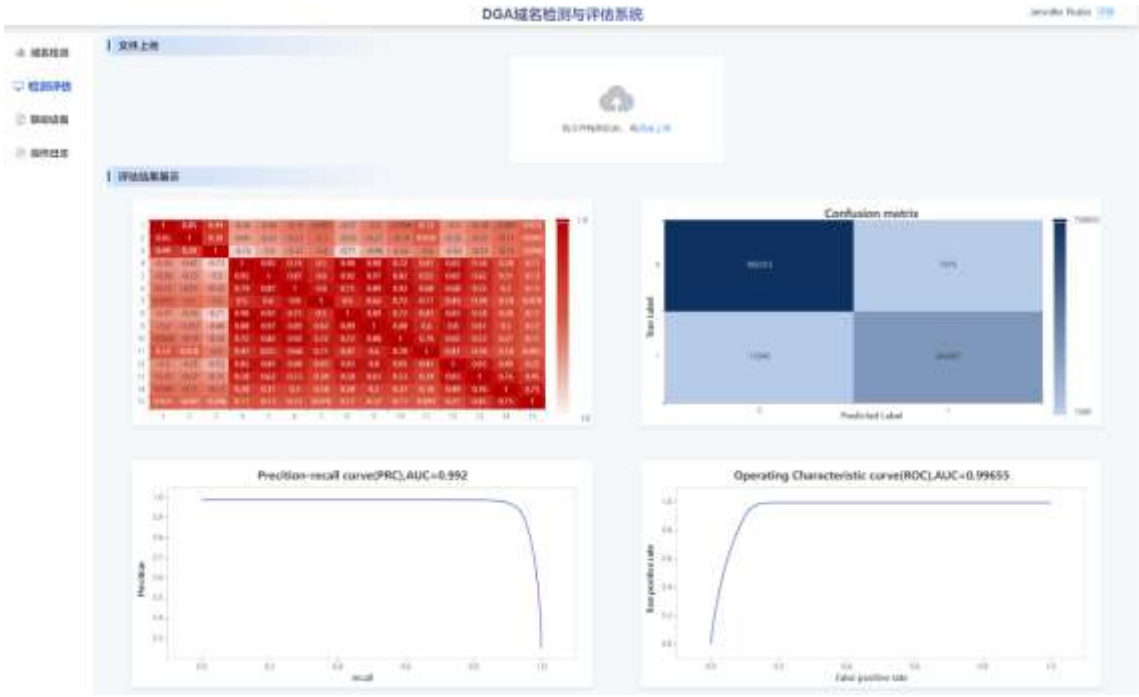

图 5-4 检测结果评估页面展示

Figure 5-4 Detection Results Evaluation Interface Display

表 5-5 联动设备配置用例

Table 5-5 Linkage Device Configuration Use Cases

测试对象	联动设备配置
测试目的	验证联动设备配置页面的功能，确保用户能够正确配置联动设备的 IP 地址、端口号、设备侧北向用户及密码，并验证连通性测试的功能。
预置条件	1. 系统已部署并运行，具备联动设备配置页面的访问权限。 2. 测试环境中已配置至少一个联动设备。
操作步骤	1. 登录系统，并访问联动设备配置页面。 2. 输入联动设备的 IP 地址和端口号。 3. 输入北向用户的用户名和密码。 4. 执行连通性测试。 5. 观察系统连通性测试是否成功，并可以配置联动设备。
预期结果	1. 系统允许用户输入联动设备的 IP 地址和端口号。 2. 系统允许用户输入北向用户的用户名和密码。 3. 系统能够成功配置联动设备。 4. 系统能够完成连通性测试，并显示测试结果。
实际结果	测试结果一致，设备配置及连通如图 5-5 所示

图 5-5 配置联动界面测试

Figure 5-5 Testing of the Configurable Linkage Interface

（3）联动设备配置测试：联动设备用于接收上报的 DNS 流量信息，在配置页面会配置设备的 IP 以及端口号，还需要输入设备的北向用户及密码，并且提供连通性测试，配置完成后采集模块将能够接收并解析对应设备上送的日志数据，测试用例如上表 5-5 所示。

图 5-6 日志功能测试

Figure 5-6 Log Function Testing

表 5-6 日志记录用例

Table 5-6 Logging Use Cases

测试对象	日志记录
测试目的	验证日志记录模块的功能，确保该模块能够准确记录用户的登录、修改配置等操作，以便于后续的审计和分析
预置条件	1. DGA 检测系统已部署并运行。 2. 系统管理员已创建至少一个测试用户账户。 3. 系统配置允许测试用户进行登录和修改配置等操作。
操作步骤	1. 登录系统，使用测试用户账户。 2. 执行至少一次用户登录操作。 3. 执行至少一次修改配置的操作，例如更改系统设置或调整检测参数。 4. 观察日志记录模块是否准确记录了上述操作。 5. 检查日志记录中的操作时间、用户信息、操作内容等信息是否完整准确。
预期结果	1. 系统允许测试用户登录，并记录了登录操作。 2. 系统允许测试用户修改配置，并记录了修改配置的操作。 3. 日志记录模块准确记录了上述操作，包括操作时间、用户信息、操作内容等信息。 4. 日志记录中的信息完整准确，没有遗漏或错误。
实际结果	测试结果一致，日志页面显示正常，如上图 5-6 所示：

（4）日志记录测试：用于记录用户操作日志，属于安全性测试，主要针对增删改信息进行记录，记录信息满足具备审计的条件，即包含操作用户、操作时间、操作内容、操作结果等内容。测试用例如表 5-6 所示。

（5）近期趋势分析测试：本部分主要涉及前端页面展示，主要包括两个部分：近期DGA恶意域名检测趋势、自定义时间过滤。此功能可以在前端页面展示趋势图，并且允许用户根据自定义时间范围，查看该段时间内系统接收到的DGA域名及其统计信息。用例如表 5-7 所示。

（6）自定义时间过滤测试：在大量数据的列表中，提供了根据自定义时间段查询DGA域名信息能力，该功能允许用户根据选择的时间范围过滤出特定时间内的 DGA 域名及其统计信息。详细测试用例如表 5-8 所示。

表 5-7 DGA 检测系统近期趋势图功能测试

Table 5-7 Functional Test of the Recent Trend Chart in the DGA Detection System

测试对象	DGA 检测系统近期趋势图功能测试
测试目的	验证 DGA 检测系统前端页面上的近期趋势图功能，确保该功能能够正确展示系统接收到的 DGA 域名随时间的变化趋势。
预置条件	1. DGA 检测系统已部署并运行。 2. 系统前端页面已正常显示，且具备访问权限。 3. 系统已成功接收至少一段时间的 DGA 域名数据。
操作步骤	1. 登录系统，并访问前端页面上的近期趋势图功能。 2. 观察系统是否正确展示了 DGA 域名随时间的变化趋势。 3. 确认趋势图是否能够清晰地反映 DGA 域名的数量变化。
预期结果	1. 系统前端页面正确展示了 DGA 域名随时间的变化趋势。 2. 趋势图能够清晰地反映 DGA 域名的数量变化。
实际结果	测试结果一致，近期趋势分析页面显示正常，如图 5-7 所示

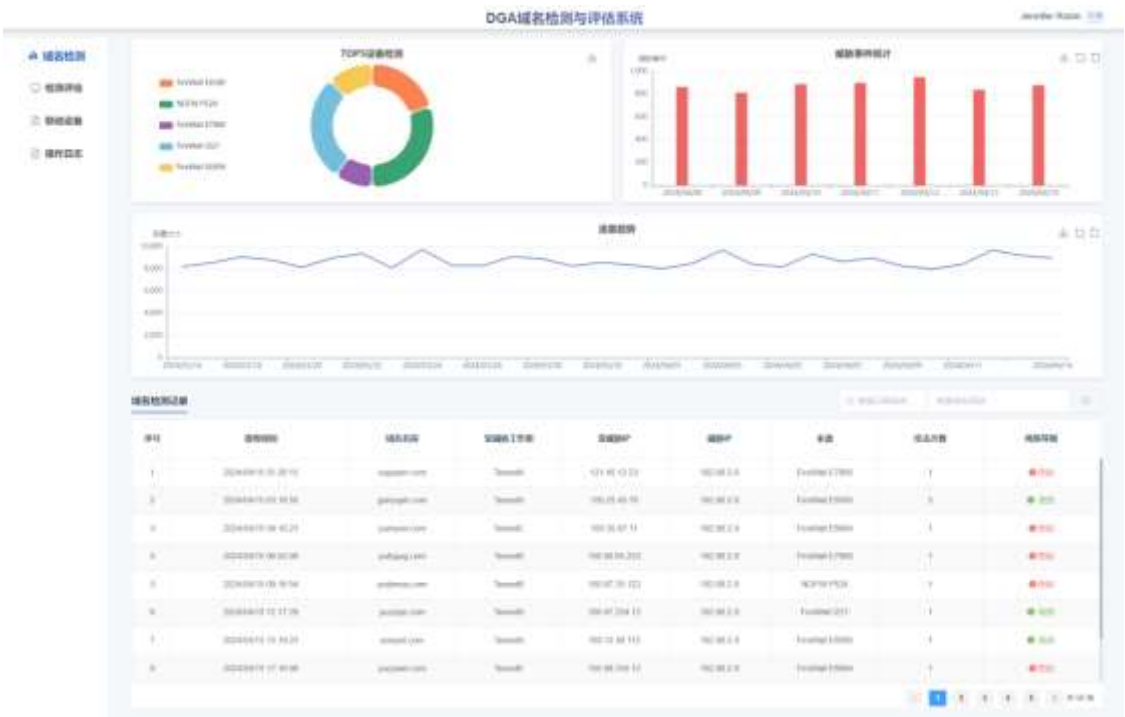

图 5-7 DGA 检测系统近期趋势图功能测试

Figure 5-7 Testing of the Recent Trends Graph Function in the DGA Detection System

表 5-8 DGA 检测系统自定义时间过滤功能测试

Table 5-8 Test of the Custom Time Filtering Function in the DGA Detection System

测试对象	DGA 检测系统自定义时间过滤功能测试
测试目的	验证 DGA 检测系统前端页面上的自定义时间过滤功能，确保该功能能够根据用户选择的条件过滤出特定时间范围内的 DGA 域名及其统计信息。
预置条件	1. DGA 检测系统已部署并运行。 2. 系统前端页面已正常显示，且具备访问权限。 3. 系统已成功接收至少一段时间的 DGA 域名数据。
操作步骤	1. 登录系统，并访问前端页面上的自定义事件过滤功能。 2. 选择一个自定义时间范围，例如过去一个月或过去一周。 3. 观察系统是否正确显示了所选时间范围内的 DGA 域名及其统计信息。 4. 确认过滤条件是否能够准确筛选出符合要求的 DGA 域名。
预期结果	1. 系统允许用户根据自定义时间范围查看 DGA 域名及其统计信息。 2. 系统前端页面正确展示了所选时间范围内的 DGA 域名及其统计信息。 3. 过滤条件能够准确筛选出符合要求的 DGA 域名。
实际结果	测试结果一致，根据自定义时间过滤域名信息如图 5-8 所示

图 5-8 威胁事件自定义时间过滤

Figure 5-8 Custom Time Filtering for Threat Events

（7）DGA 恶意域名情报查看测试：查询页面用户可以输入域名并进行查询，系统会与威胁情报库中的信息进行模糊匹配，若存在域名信息，则显示域名详情，若不存在则显示暂无数据，用例如表 5-9。

表 5-9 DGA 恶意域名情报查看用例

Table 5-9 Use Case for Viewing DGA Malicious Domain Intelligence

测试对象	DGA 恶意域名情报查看
测试目的	本测试用例旨在验证恶意域名情报查看功能是否能够准确收录当前已知的各类 DGA 及其相应的域名详细信息，并确保该功能能够正确响应用户的查询请求。
预置条件	1. 查询页面已部署并运行。 2. 威胁情报库已包含至少一个已知的 DGA 域名及其详细信息。 3. 测试用户具备访问 DGA 恶意域名情报查询页面的权限。
操作步骤	1. 登录系统，并访问 DGA 恶意域名情报查询页面。 2. 在查询框中输入一个已知的 DGA 域名。 3. 观察系统是否能够正确响应查询请求。 4. 确认系统是否能够与威胁情报库中的信息进行匹配。 5. 检查系统是否能够显示查询结果，包括域名详情信息。
预期结果	1. 系统允许用户输入查询域名，并正确响应查询请求。 2. 系统能够与威胁情报库中的信息进行匹配。 3. 系统能够显示查询结果，包括域名详情信息。
实际结果	测试结果一致，查询测试如图 5-9 所示

图 5-9 DGA 恶意域名情报查看测试

Figure 5-9 Custom Time Filtering for Threat Events

（8）系统可靠性测试：针对异常场景，测试系统恢复能力，用例如表 5-10：

表 5-10 系统可靠性用例

Table 5-10 System Reliability Use Cases

测试对象	系统可靠性测试
测试目的	验证 DGA 检测系统的可靠性，确保系统能够在各种异常情况下保持稳定运行，业务不中断。
预置条件	1. 系统已部署并运行在测试环境中。 2. 测试环境中已配置至少一个联动设备。
操作步骤	1. 对整个服务器进行下电重启，等待一段时间后，重新启动服务器。 2. 登录系统，观察业务是否恢复正常。 3. 在服务器网卡上注入小于 3 秒的延迟。登录系统，观察业务是否受到影响。 4. 关闭联动设备导致联动设备下线后，登录系统，观察前端页面是否显示友好提示。
预期结果	1. 单服务器下电重启后，系统能够启动并恢复正常运行，业务中断时长小于最小 SLA。 2. 系统注入网卡延迟小于 3 秒时，业务不受影响，系统能够正常处理数据。 3. 联动设备下线后，前端页面能够友好提示设备离线，并提供相应的操作建议。
实际结果	测试结果一致

5.5 本章小结

在本章中，本文对域名检测与评估系统进行了全面的设计与实现。首先，针对系统的需求进行了深入的分析和梳理，包括应用需求、部署需求、功能需求以及非功能需求。基于对需求的分析，进行了系统的总体架构设计。基于制定的目标场景，实现了基于大数据开源生态系统组件的DGA恶意域名检测与评估系统。最后，从功能、可靠性、安全性等多个角度对系统进行了全面的测试。这些测试不仅验证了系统的基本功能，也确保了系统可以长期稳定的提供服务。

6 总结与展望

6.1 本文总结

随着互联网的发展，网络已经成为人们生活中不可或缺的一部分，但是在网络带给本文遍历的同时，也潜藏着巨大的威胁，在网络安全问题中，僵尸网络是目前较为因为瞩目的一类。而它会通过利用DGA恶意域名实现对内网主机的命令控制或数据窃取。现在已经有不少具有良好效果的DGA检测手段，但是检测周期较长，且缺少可靠性。并且现有的评估指标大多是对算法的效果说明，缺少反馈机制，且鉴于本系统采用线下部署形式。为解决上述问题，本文研制了一套运行在轻量级系统上的检测与评估系统，可以针对现网中的流量联动设备，实现对流量及日志的DGA检测，并且系统中的算法模块采取组件式的设计思想，能够实现其他恶意流量或行为检测能力的快速扩展，系统前后端分布采用 Vue 框架,后端采用主流开发语言 Python,通过编程实现了基于机器学习的 DGA 恶意域名检测系统,该系统能够对相应攻击进行识别和统计,具备较好的精确性和可靠性,系统易于维护和扩展,达到了预期效果。本系统基于机器学习算法，实现了 DGA 恶意域名的快速实时监测，且误报低率，系统鲁棒性高。因此对DGA检测具有一定意义。本文的主要工作如下：

（1）提出了一种基于机器学习的 DGA 域名检测方法。引入了两阶段检测流程。首先利用聚类算法对具有相似性的DGA域名进行聚类，有效发现和挖掘这些域名之间的内在联系和规律。其次，考虑到DGA域名本身具有的随机性，采用随机森林分类算法对聚类结果进行进一步的判断和分类，以提高分类的准确性和稳定性。实验结果表明，本文提供的检测方法在误报率和检测效率上有较好的表现，适用于上述目标场景。

（2）提出了一种 DGA 检测评估框架。现有的评估指标大多是对算法的效果说明，缺少反馈机制，且本系统采用线下部署形式难以持续更新。为解决上述问题，本系统设计了评估框架并引入运营评估中心。该中心依据系统在实际应用场景中运营人员所关注的特定指标，构建了一套评估体系，实时更新评估数据，反馈给检测模型做持续优化。该框架通过多组数据集验证，可以帮助运营人员定期优化检测效果并动态展示评估结果，确保其在实际应用中的有效性和可靠性。

（3）设计并开发了基于大数据平台的 DGA 检测与评估系统。充分分析需求后，本文构建的恶意域名检测系统可以对接现网中的防火墙设备，针对边界设备上报的DNS日志以及网卡流量进行分析，检测出恶意域名会上报前台展示并对结果进

行评估，维护一个健康的网络环境

（4） 开展了系统实现和测试。本文所实现的恶意域名检测系统建立在分布式大数据平台上,对系统的软硬件环境进行配置,并根据广泛应用的大数据开源生态的 Hadoop、Spark 等组件搭建了系统集群的运行环境和开发环境,并利用这些大数据技术对数据采集模块、检测与评估模块、存储模块、可视化模块等进行了实现。在核心检测模块中，利用 Spark 大数据技术较好地适应了海量网络流量日志的检测,无论是分类模型的建立还是对现网中的流量日志恶意域名的检测都具有较高的效率与可靠性。采用可扩展的组件开发技术为后续进一步提升系统的检测范围提供扩展能力。最后采用科学测试方法，对系统进行功能测试和可靠性测试,验证了系统功能的完整性、正确性与算法的可靠性。

6.2 未来展望

本文采用的机器学习领域的聚合与分类方法，对DGA恶意域名的检测与评估展开了研究，并详细介绍了整体系统的设计和实现过程。但目前本文的研究系统仍然存在不足和需要改进之处，需要在未来的工作中进一步改进和研究，主要包括以下几个方面：

（1）系统的目标场景中，对低误报率的要求较高，因此，在模型训练过程中，为了降低误报率，导致在漏报率上有所欠缺。未来的研究可以探讨如何进一步降低漏报率，这需要对相关文献进行深入研究，拓宽理论基础和方法论，并设计对照组，以增强模型的稳定性和可靠性。

（2）评估模型的评估指标采用传统方法，尚未进行深入研究。后续研究可以针对系统的具体应用场景，对评估框架进行优化，以提高评估的准确性和实用性。

（3）系统性能有待进一步优化。尽管当前系统已实现完整功能，但在某些模块中仍有进一步优化的空间。例如，可以通过引入缓存技术如 Redis 来提升系统对情报库的读取性能，或者对 Spark 资源分配策略进行进一步优化，以提高系统的整体性能。

（4）域名行为场景识别的算法集成不足。目前系统中主要实现了 DGA 域名检测和域名统计分析功能。未来研究可以将更多类型的域名行为识别算法，如钓鱼域名检测、DNS 隧道检测等，集成到系统中，从而完善域名行为观测系统的检测能力。

参考文献

[1] Štrbová M, Kuzior P. Safety Management in the Age of Internet Threats[J]. Management Systems in Production Engineering, 2019, 27(2): 88-92.

[2] Alessandro C , Christian M , Luca S , et al. Algorithmically Generated Malicious Domain Names Detection Based on n-Grams Features[J]. Expert Systems with Applications, 2020, (prepublish): 30-32.

[3] Anderson S H, Woodbridge J, Filar B. DeepDGA: Adversarially-Tuned Domain Generation and Detection. [J]. CoRR, 2016, abs/1610.01969: 14-15.

[4] Xiaochun Y, Ji H, Yipeng W, et al.Khaos: An Adversarial Neural Network DGA With High Anti-Detection Ability[J]. IEEE Transactions on Information Forensics and Security, 2020, 152225-2240:54-57.

[5] Satoh A, Nakamura Y, Nobayashi D, et al. Estimating the Randomness of Domain Names for DGA Bot Callbacks[J]. IEEE Communications Letters, 2018:1-1.

[6] 于光喜，张桉，崔华俊，杨兴华，李杨，刘畅．基于机器学习的僵尸网络 DGA 域名检测系统设计与实现[J] ．信息安全学报，2020, 5(03):35-47.

[7] Leyla, Bilge, Sevil, et al. Exposure: A Passive DNS Analysis Service to Detect and Report Malicious Domains[J]. ACM Trans. Inf. Syst. Secur, 2014, 16(4): 14:1-14:28.

[8] Truong D T, Cheng G. Detecting domain‐flux botnet based on DNS traffic features in managed network[J]. Security and Communication Networks, 2016, 9(14): 2338-2347.

[9] Davuth N, Kim S R. Classification of Malicious Domain Names using Support Vector Machine and Bi-gram Method[J]. International Journal of Security & Its Applications, 2013: 7.

[10] Woodbridge J, Anderson S, Ahuja A, et al. Predicting domain generation algorithms with long short-term memory networks[J]. arXiv preprint arXiv, 2016, 1611.00791: 9-11.

[11] Anderson S, Woodbridge J, Filar B. DeepD GA: Adversarially-tuned domain generation and detection[C]. Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security, New York, USA, 2016: 13-21.

[12] Yin C, Zhu Y, Liu S, et al. An enhancing framework for botnet detection using generative adversarial networks[C]. 2018 International Conference on Artificial Intelligence and Big Data, Chengdu, China, 2018: 228-234.

[13] Yu B, Gray D L, Pan J, et al. Inline DGA detection with deep networks[C]. 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, USA, 2017: 683-692.

[14] Mac H, Tran D, Tong V, et al. DGA botnet detection using supervised learning methods[C].

Proceedings of the 8th International Symposium on Information and Communication Technology, Nha Trang City, Viet Nam, 2017: 211-218.

[15] Vinayakumar R, Soman K P, Poornachandran P, et al. Evaluating deep learning approaches to characterize and classify the DGAs at scale[J]. Journal of Intelligent and Fuzzy Systems, 2018, 34(3): 1265-1276.

[16] Zhou C, Sun C, Liu Z, et al. A C-LSTM neural network for text classification[J]. Expert Systems with Applications, ELSEVIER, 2017, 72: 221-230.

[17] Pereira M, Coleman S, Yu B, et al. Dictionary extraction and detection of algorithmically generated domain names in passive DNS traffic[C]. International Symposium on Research in Attacks, Intrusions, and Defenses. Springer, Cham, 2018: 295-314.

[18] Yang L, Liu G, Zhai J, et al. A novel detection method for word-based DGA[C]. International Conference on Cloud Computing and Security. Springer, Cham, 2018: 472-483.

[19] Meher A ,Muntaka I ,Ashikur R , et al.On Feature Selection Algorithms for Effective Botnet Detection[J].Journal of Network and Systems Management,2024,32(2): 56-60.

[20] 李可，方滨兴，崔翔等．僵尸网络发展研究[J]．计算机研究与发展，2016, 53(10): 2189-2206.

[21] Hoque N, Bhattacharyya D K, Kalita J K. Botnet in DDoS Attacks: Trends and Challenges[J]. IEEE Communications Surveys & Tutorials, 2015, 17(4): 2242-2270.

[22] JS B ,Sehgal ,Kumar S .Botnet Command Detection using Virtual Honeynet[J].International Journal of Network Security Its Applications,2011,3(5): 177-189.

[23] Tong A T, Long H V, Taniar D. On Detecting and Classifying DGA Botnets and their Families - ScienceDirect[J].2021: 54-55.

[24] Xuan D C ,Duong V L ,Nikolaevich V T .Detecting CC Server in the APT Attack based on Network Traffic using Machine Learning[J].International Journal of Advanced Computer Science and Applications (IJACSA),2020,11(5): 65-66.

[25] Sun Y Q, Jian K L, Cui L Y, et al. Online malicious domain name detection with partial labels for large-scale dependable systems[J]. Journal of Systems and Software, 2022: 34-40.

[26] Patsakis C, Casino F. Exploiting statistical and structural features for the detection of Domain Generation Algorithms[J]. Journal of Information Security and Applications, 2021, 58(2): 6576-6589.

[27] Filipa R ,Miguel M D G ,Pedro F .Controlling digital piracy via domain name system blocks: A natural experiment[J].Journal of Economic Behavior and Organization,2024,21889-103:65-67.

[28] Symantec Corporation. Patent Issued for Techniques for Avoiding Dynamic Domain Name System (DNS) Collisions (USPTO 9130994)[J]. Telecommunications Weekly, 2015 :55-56

[29]王浩．基于机器学习的异常 DNS 流量检测研究[D]．南京邮电大学，2020. 45-47.

[30] Yuchen L, Yanan C, Zhaoxin Z, et al.Illegal Domain Name Generation Algorithm Based on Character Similarity of Domain Name Structure[J]. Applied Sciences, 2023, 13(6): 4061-4061.

[31] Fu Y, 0001 Y L, Hambolu O, et al. Stealthy Domain Generation Algorithms[J]. IEEE Trans. Information Forensics and Security, 2017,12(6):1430-1443.

[32] Constantinos P, Fran C. Exploiting statistical and structural features for the detection of Domain Generation Algorithms[J]. Journal of Information Security and Applications, 2021, 58: 44-56

[33] Yu B, Pan J, Gray L D, et al. Weakly Supervised Deep Learning for the Detection of Domain Generation Algorithms[J]. IEEE Access, 2019, 751542-51556: 54-65.

[34] Engineering - Materials Engineering; Study Data from Hefei University of Technology Provide New Insights into Materials Engineering (Detecting Domain Generation Algorithms with Bi-LSTM)[J]. Mathematics Week, 2020, 12(5): 1530-1543.

[35] SPEISER J L, MILLER M E, TOOZE J, et al. A comparison of random forest variable selection methods for classification prediction modeling[J]. Expert systems with applications, 2019, 134: 93-101.

[36] 王奕森，夏树涛．集成学习之随机森林算法综述[J]．信息通信技术，2018, 12(01)：49-55.

[37] KIM H, LIM Y. Bootstrap aggregated classification for sparse functional data[J]. Journal of Applied Statistics, 2022, 49(8): 52-63.

[38] HOROWITZ J L. Bootstrap methods in econometrics[J]. arXiv preprint arXiv, 2018, 180904016:65-66.

[39] Lee T G, Lim G H, Wang T, et al. Double bagging trees with weighted sampling for predictive maintenance and management of etching equipment[J]. Journal of Process Control, 2024, 135103175: 35-37.

[40] SALEENA N. An ensemble classification system for twitter sentiment analysis [J]. Procedia computer science, 2018, 132: 37-46.

[41] Dau X H, Hanh X V. An improved model for detecting DGA botnets using random forest algorithm[J]. Information Security Journal: A Global Perspective, 2022, 31(4): 441-450.

[42] LOH W-Y. Classification and regression trees [J]. Wiley Interdisciplinary Reviews-Data Mining and Knowledge Discovery, 2011, 1(1): 14-23.

[43] Ali S, Fatemeh A, Saman C S. A Novel Approach for Detecting DGA-Based Botnets in DNS Queries Using Machine Learning Techniques[J]. Journal of Computer Networks and Communication s, 2021, 2021:76-79

[44] H. S. Anderson, J. Woodbridge, B. Filar. DeepD GA: Adversari-ally-Tuned Domain Generation and Detection[C]. the 2016 ACM Workshop on Artificial Intelligence and Security (AISec), 2016. 13-21.

[45] Alessandro C, Christian M, Luca S, et al. Algorithmically Generated Malicious Domain Names Detection Based on n-Grams Features[J]. Expert Systems with Applications, 2020, (prepublish): 451-456.

[46] Yuchen L, Yanan C, Zhaoxin Z, et al. Illegal Domain Name Generation Algorithm Based on Character Similarity of Domain Name Structure[J]. Applied Sciences, 2023, 13(6): 4061-4061.

[47] Alexa Internet, Inc. Does Alexa have a list of its top-ranked websites[EB/OL]. [2018-09-03]. https://support.alexa.com/hc/en-us/articles/200449834-Does-Alexa-have-a-list-of-its-top-ranked-websites-.

[48] Pochat V L, Van Goethem T, Tajalizadehkhoob S, et al. Tranco: A research-oriented top sites ranking hardened against manipulation[EB/OL]. (2018-12-17)[2022-11-22]. https://arxiv.org/abs/1806.01156

[49] Qihoo 360 Technology Co, Ltd. 360 netlab OpenD ata project DGA feeds [EB/OL]. [2018-09-03]. https://data.netlab.360.com/dga/.

[50]ABAKUMOVA. andrewaeva/DGA[CP/OL]. (2022-09-15). https://github.com/andrewa eva/DGA

[51] 韩春雨，张永铮，张玉．Fast-flucos．基于 DNS 流量的 Fast-flux 恶意域名检测方法[J]．通信学报，2020, 41(05): 37-47.

[52] 周江，王伟平，孟丹等．面向大数据分析的分布式文件系统关键技术[J]．计算机研究与发展，2014, 51(2): 382-394.

学位论文数据集

表 1.1： 数据集页

关键词*	密级*	中图分类号	UDC	论文资助
恶意域名;机器学习;随机森林;大数据;	公开			
学位授予单位名称*		学位授予单位代码*	学位类别*	学位级别*
北京交通大学		10004	电子信息	硕士
论文题名*		并列题名*		论文语种*
DGA 恶意域名检测与评估系统的研究与实现		无		中文
作者姓名*	李美慧		学号*	22140488
培养单位名称*		培养单位代码*	培养单位地址	邮编
北京交通大学		10004	北京市海淀区西直门外上园村 3 号	100044
专业学位类别（领域）*		研究方向*	学制*	学位授予年*
软件工程		网络安全	两年	2024
论文提交日期*	2024 年 5 月			
导师姓名*	常晓林		职称*	教授
评阅人	答辩委员会主席*		答辩委员会成员	
	张宝鹏		吴丹 韩文娟	
电子版论文提交格式　文本（ ）　图像（ ） 视频（ ） 音频（ ） 多媒体（ ） 其他（ ） 推荐格式：application/msword；application/pdf				
电子版论文出版（发布）者		电子版论文出版（发布）地		权限声明
论文总页数*	77			
共 33 项，其中带*为必填数据，为 21 项。				