分类号：_____ 密级：_____

U D C：_____ 编号：_____

专业学位硕士学位论文

基于深度学习的 DGA 域名检测方法研究

硕士研究生 ：王海茹

指 导 教 师 ：郭方方 副教授

校 外 导 师 ：贺 劼 高级工程师

学 位 类 别 ：工程硕士

哈尔滨工程大学

2024 年 6 月

分类号：_____ 密级：_____

U D C ：_____ 编号：_____

专业学位硕士学位论文

基于深度学习的 DGA 域名检测方法研究

（□产品研发□工程规划□工程设计☑应用研究）

硕士研究生 ：王海茹

指 导 教 师 ：郭方方　副教授

校 外 导 师 ：贺　劼　高级工程师

专 业 类 别 ：软件工程

所 在 学 院 ：计算机科学与技术学院

论文提交日期 ：2024 年 04 月

论文答辩日期 ：2024 年 05 月

学位授予单位 ：哈尔滨工程大学

Classified Index:
U.D.C:

A Thesis for the Degree of Master of Engineering

Research on DGA Detection Method Based on Deep Learning

Candidate: Wang Hairu
Supervisor: A.P. Guo Fangfang
Associate Supervisor: Senior Engineer He Jie
Professional category: Software Engineering
College: Computer Science and Technology
Date of Submission: April, 2024
Date of Oral Examination: May, 2024
University: Harbin Engineering University

摘 要

随着互联网在人们生活中的普及和深入，网络安全问题逐渐凸显。僵尸网络（Botnet）是互联网信息系统的一个常见且严重的威胁。它由大量被黑客通过恶意软件控制的计算机组成，这些计算机可以在黑客的指挥下执行各种恶意活动。因此僵尸网络的检测和防范在网络安全中十分重要。域生成算法（Domain Generation Algorithm，DGA）域名检测是一种重要的僵尸网络检测技术，这一方法仅需分析域名字符而不依赖其他信息，易于在网络各节点实施，是现在僵尸网络检测的主流技术。

本文对目前 DGA 域名检测的主要方法进行了深入的检索和分析，并探讨了 DGA 域名检测所面临的几个关键性问题。针对这些问题进行了算法改进优化的研究，主要研究内容如下：

（1）针对部分 DGA 域名家族样本稀缺性问题，设计了一个基于生成对抗网络（Generative Adversarial Network，GAN）的 DGA 域名生成算法。该算法利用 n-gram 字典和 WGAN（Wasserstein GAN）相结合的方法来优化域名生成模型，在保证模型稳定收敛的同时，生成与目标相似的域名。该方法有助于增强小样本 DGA 域名家族训练集的质量。

（2）针对短域名和基于单词表的 DGA 域名检测效果差的问题，设计了一个基于域名长度的异构的 DGA 域名检测模型，该模型将不同的特征工程和深度神经网络集成在一个统一的检测方法中。对于字符级域名，采用了注意力递归图（Attentional Recurrence Plot，ARP）的方式进行数据处理，设计了并行多路卷积模块进行特征提取，在短域名的检测上取得了显著的效果。对于单词级域名，采用 n-gram 提取特征方法，并利用门控循环单元（Gate Recurrent Unit，GRU）结合注意力，使模型不仅提升了对数据中关键信息的捕捉能力，还增强了对 DGA 域名长期依赖关系的理解，从而提升了基于单词表的域名检测能力。

实验证明，应用本文中设计的样本生成方法所产生的数据，在特征上与实际的恶性域名紧密匹配，这有效验证了此方法在扩充恶意域名数据集上的实用性。此外，本文提出的检测模型与其他模型相比，误报率更低，准确率更高，展现出其优越性。

关键词：僵尸网络；DGA 域名；生成对抗网络；双向长短期记忆神经网络；域名检测

Abstract

As the internet becomes increasingly integrated into people's lives, the issue of cybersecurity is becoming more prominent. Botnets represent a common and severe threat to internet information systems. A botnet is composed of a large number of computers controlled by hackers through malicious software. These computers can carry out various malicious activities under the command of the hackers. Therefore, the detection and prevention of botnets are critical in the realm of cybersecurity. Domain Generation Algorithm (DGA) domain name detection is a key technique for detecting botnets. This method only requires the analysis of domain name characters and does not depend on other information, making it easy to implement across various network nodes and currently a mainstream technology for botnet detection.

This thesis thoroughly searches and analyzes the current main methods for detecting DGA domain names and delves into several key issues faced by DGA domain name detection. In response to these issues, research has been conducted to improve and optimize algorithms, with the main content of the study as follows:

(1) To address the issue of scarce samples in some DGA domain name families, a DGA domain name generation algorithm based on Generative Adversarial Networks (GAN) has been developed. This algorithm employs a combination of n-gram dictionaries and Wasserstein GAN (WGAN) to optimize the domain name generation model. By ensuring stable convergence of the model, it generates domain names that are similar to the target. This method helps to enhance the quality of the training sets for small-sample DGA domain name families.

(2) To address the thesis detection performance of short and wordlist-based DGA domain names, a heterogeneous DGA domain name detection model based on domain length has been designed. This model integrates various feature engineering techniques and deep neural networks into a unified detection method. For character-level domain names, data processing is carried out using the Attentional Recurrence Plot (ARP) method, and a parallel multi-path convolutional module is designed for feature extraction, achieving significant results in the detection of short domain names. For word-level domain names, an n-gram feature extraction method is used, combined with

Gated Recurrent Units (GRU) and attention mechanisms. This approach not only enhances the model's ability to capture key information within the data but also improves its understanding of the long-term dependencies of DGA domain names, thereby enhancing the detection capability for wordlist-based domain names.

Experiments demonstrate that the data produced by the sample generation method designed in this paper closely matches the features of actual malicious domain names, effectively validating the practicality of this method in expanding malicious domain name datasets. Moreover, the detection model proposed in this article has a lower false positive rate and higher accuracy compared to other models, showcasing its superiority.

Keywords: Botnet, DGA domain, Generative Adversarial Network, BiLSTM, Domain Name Detection

目 录

第 1 章 绪论.....1
1.1 研究背景及意义.....1
1.2 国内外研究现状.....2
1.2.1 基于生成对抗网络的数据增强技术.....2
1.2.2 基于机器学习的 DGA 域名检测技术.....3
1.2.3 基于深度学习的 DGA 域名检测技术.....4
1.3 论文主要内容.....5
1.4 论文组织结构.....6
第 2 章 基于生成对抗网络的 DGA 域名样本生成.....9
2.1 问题的提出.....9
2.2 域名分析与序列学习.....9
2.2.1 n-gram 分析域名原理.....9
2.2.2 BiLSTM 处理域名原理.....11
2.3 基于 GAN 的 DGA 域名生成模型.....13
2.3.1 模型概述.....13
2.3.2 n-gram 生成器.....14
2.3.3 域名处理器.....14
2.3.4 域名生成器.....15
2.3.5 用 NWGAN 生成数据.....20
2.4 本章小结.....21
第 3 章 异构的 DGA 域名检测模型.....23
3.1 问题的提出.....23
3.2 域名检测技术.....24
3.2.1 卷积神经网络与域名数据分析.....24
3.2.2 递归图与域名动态分析.....25
3.2.3 注意力机制与域名特征处理.....26
3.3 异构 DGA 域名检测模型.....28
3.3.1 总体模型.....28
3.3.2 域名字符处理规则.....29
3.3.3 字符级域名检测模块.....31
3.3.4 单词级域名检测模块.....34
3.4 本章小结.....37
第 4 章 实验结果与分析.....39
4.1 基于 GAN 的 DGA 域名生成方法实验.....39

4.1.1 实验环境和实验设计.....39
4.1.2 数据集.....40
4.1.3 模型评价指标.....40
4.1.4 生成域名分析.....41
4.1.5 生成域名评估.....42
4.1.6 对比实验.....46
4.2 异构 DGA 域名检测模型方法实验.....47
4.2.1 实验设计.....47
4.2.2 数据集.....48
4.2.3 模型评价指标.....49
4.2.4 ARP 消融实验.....50
4.2.5 二分类实验结果.....51
4.2.6 多分类实验结果.....54
4.3 本章小结.....59
结论.....61
参考文献.....63

第 1 章 绪论

1.1 研究背景及意义

随着现代网络技术的发展和进步，网络与人们之间的距离越来越近。各种网络安全问题也随之而来。僵尸网络、钓鱼网站、勒索软件等恶意攻击对网络活动的正常进行构成了相当大的威胁，已成为网络安全领域的研究热点。

在这些恶意活动中，僵尸网络是由大量计算机组成的恶意攻击平台。这些计算机受到病毒、蠕虫、特洛伊木马等恶意软件的感染和控制[1]，它们遵循僵尸管理员（botmaster）的命令，botmaster 通过命令与控制（Command and Control，C&C）通道对被感染主机进行一对多的控制[2]。通过利用僵尸网络，僵尸管理员将更容易进行各种恶意活动，如发送垃圾邮件，DDOS 攻击，窃取数据，进行欺诈活动等。随着时间的推移，僵尸网络在渗透技巧和伪装方法上不断进化，对网络安全专家和系统管理员构成了持续的挑战。

DGA 僵尸网络是指部署在客户机-服务器架构下的僵尸网络。机器人充当客户端。在感染受害者的计算机后，它会连接到 C&C 服务器接收命令。他们通常通过域名系统（Domain Name System，DNS）查询查找 C&C 服务器的 IP 地址。根据自动域名生成算法，这些域名不断被更改，以绕过安全系统[3]。DGA 域名的特点是数量庞大、变化迅速和生命周期短暂，传统的基于黑名单的检测方法效力下降。基于机器学习和深度学习技术的 DGA 域名检测方法近期已被广泛探索。对于基于机器学习的 DGA 域名检测方法，其特征提取需要复杂的人工分析且耗时，攻击者可以轻易绕过这些提取的特征。这种检测方法的局限性导致了深度学习方法的兴起。现有的基于深度学习的 DGA 域名检测器在整体检测水平和大型 DGA 域名家族检测上实现了较好的分类性能。然而，近年来新的 DGA 域名变体不断涌现，跨 DGA 域名家族存在严重的数据不平衡。对于那些训练样本不足的 DGA 域名和新兴的 DGA 变体域名，现有 DGA 域名检测器的准确性显著降低。

检测 DGA 僵尸网络的核心问题在于如何有效区分恶意域名与良性域名。这个问题可以进一步转化为一个多类分类问题，目的是准确识别并分类不同僵尸网络生成的域名族。本文采用基于 DNS 分析的僵尸网络检测方法，对恶意 DGA 域名进行检测和分类。这种方法的优点是可以防止僵尸网络与 C&C 服务器通信。

通过这种方式，即使僵尸网络已经感染了计算机，也可以有效地禁用它们。此外，与其他解决方案相比，DNS 分析所需的计算资源较少，使其成为一种高效的检测方法。

1.2 国内外研究现状

1.2.1 基于生成对抗网络的数据增强技术

生成对抗网络是 2014 年由 Ian Goodfellow[4]等人提出的。主要用于图像生成、图像修复、风格迁移、艺术图像创造等任务。它包含两个在零和博弈中相互竞争的神经网络。生成器（Generator）合成看似来自期望分布的样本。判别器（Discriminator）负责区分合成样本和真实样本。

生成对抗网络被大量研究用于数据增强，Tanaka[5]等人使用 GAN 为机器学习任务生成人工训练数据，并在不同网络架构的基准数据集上进行了测试，生成的数据可以达到跟原始数据相同甚至更好的检测效果。Ren[6]等人通过提供多样化的训练样本来增强生成样本的多样性，并使用幅频域和皮尔逊相关系数来评估样本的可靠性。还建立了相频域生成模型和时域判别器模型来提高样本相似性，并避免梯度消失问题。实验证明这种方法能有效提高故障诊断性能。Klopries[7]等人提出了两种可解释的方法来生成合成数据：操纵深度自动编码器的功能来对合成数据进行采样；利用 GAN 来生成合成数据。该方法在源和目标数据集域中呈现了时间序列的可解释和明确分离的特征，与现有方法相比，具有定性和定量优势。

近年来，使用 GAN 及其变体模型进行数据增强的研究越来越多，并因此在域名检测任务中展现出巨大的潜力。傅伟[8]等人提出了一种创新的结合 skip-gram 和 WGAN 模型来生成伪 DGA 域名的方法。首先利用 skip-gram 模型对域名进行高效转换，将其转化为适合 WGAN 处理的向量表示。随后，WGAN 模型深入挖掘这些向量表示中的潜在特征，学习并生成伪 DGA 域名。这一研究为提升域名检测算法的准确性和泛化性能提供了新的思路和方法。Yu[9]等人中提出了三种用于域名生成的 GAN 变体，应用了自动编码器架构于最小二乘 GAN（LSGAN）、带梯度惩罚的 WGAN（WGAN-GP）和原始 GAN 的变体。这些生成域名在特性上与可用的良性域名相似，具有欺诈性。实验结果表明，WGAN-GP 在所有评估标准上都表现最佳。Ren[10]等人提出了一种新的模型 CL-GAN：一种基于 GAN 的 DGA 域名生成和检测的持续学习模型，它由三个部分

组成：一个以提示噪声作为输入来学习 DGA 的生成器，一个检测 DGA 的鉴别器和一个提供现有知识的教师。实验结果表明 CL-GAN 比其他 DGAs 可以产生更多的真实 DGA 域名，并且具有更强的 DGA 检测能力。

综上所述，生成对抗网络在数据增强方向已经取得了显著成果。通过生成高质量的变体数据用于扩充现有数据集，既解决了恶意域名数据收集困难的问题，又可以通过生成的高质量数据提升模型对于未知 DGA 域名的识别程度。

1.2.2 基于机器学习的 DGA 域名检测技术

基于机器学习的检测与分类是指使用传统机器学习中的算法和先验知识来构建特征数据集或计算数据之间的各种潜在数学关联，以对各种数据进行分类。DGA 域名通常由特定算法生成，与正常域名相比，它们在多方面存在显著差异。因此，合适的特征提取是使用机器学习进行 DGA 域名检测的关键步骤。

首先基于监督学习的特征提取与检测，Huang[11]等人提出一种基于 SVM 的 GWO 下 DGA 域名检测模型。它使用 GWO 对 SVM 的参数进行优化，提高最优参数的搜索速度。实验表明，随着计算速度的加快和精度的提高，算法的性能得到了显著提高。Mu[12]等人提出了一种方法。首先使用提取的文本特征对捕获的域进行预过滤，然后使用不同的机器学习算法和提取的 n-gram 特征来训练和评估系统。结果表明，所提方法能够捕获可疑数据包并准确分类域。

尽管基于监督学习的方法在 DGA 域名分类方面已经展现出显著的效果，但这些方法往往是针对特定数据集设计的，高度依赖于基于先验知识的特征工程，这无疑增加了分析的复杂性，泛化能力受限，容易出现过拟合现象。为了缓解监督学习模型工作量大、容易过拟合的问题，提高模型的泛化能力。基于无监督学习的 DGA 域名分类方法已经开始出现。

Moubayed[13]等人提出了一种基于机器学习的 DNS 漏洞处理方法。这是一种基于多数投票的集成学习分类器，可以检测可疑的域名。最后通过使用无监督机器学习聚类算法对未标记的数据集中的相同特征进行研究，并应用所开发的集成学习分类器进行验证。实验结果表明，所开发的集成学习分类器在保持较高查全率的同时，在准确率、精密度和 F-score 方面都有较好的表现。Park[14]等人提出了一个新的无监督学习恶意域检测模型。仅使用模型训练期间提供的良性域的自编码器，可以用更少的标记工作构建恶意域检测模型。实验结果表明，所提出的恶意域检测模型实现了 99%准确率和 F1 分数的精确检测性能。但是该

方法只能针对特定类型的 DGA 域名。

虽然基于无监督机器学习的 DGA 域名分类方法降低了数据特征提取的难度，具有较好的适用性，但仍存在一些不足。针对新型变体域名不能有效识别。机器学习算法的超参数调整多基于经验，这可能导致算法训练的难度增加，分类结果的波动较大。

1.2.3 基于深度学习的 DGA 域名检测技术

基于深度学习的 DGA 域名检测方法利用神经网络的自拟合性，能够规避传统机器学习繁杂的特征工程。通过字符嵌入、单词嵌入等方法，将域名字符转换成包含一定语法信息的特征向量。此外，模型分类性能也优于传统机器学习算法。

在基于深度学习的方法中，Tran[15]等人提出了一个名为 LSTM.MI 的新模型，继承了传统长短期神经网络（long short term memory，LSTM）的优点，并对其进行了强化，以最大限度地提高准确率并最小化噪声。测试结果显示，该算法比传统 LSTM 模型的准确率提高了至少 7%。然而，这个实验只用了小规模的数据集，不具有普遍性。Curtin[16]等人提出了一个名为 Smashword 量表的新概念，用于衡量 DGA 域家族和良性域之间的相似性。然后利用递归神经网络和侧信息网络模型来应用上述测量标准，提高了模型的检测精度。此外，Simran[17]等人采用了几种基于 n-gram 的特征表示技术来建模。对比结果显示，CNN-LSTM 模型在二元分类问题上取得了最有效的性能，其 F1 得分达到了 96.3%。

为了增强对日益变化的新型 DGA 域名的检测能力，研究人员使用额外的策略提高检测率，取得了不错的效果。Liu[18]等人提出一种基于 LSTM-CapsNet 的序列胶囊网络。该模型使用双向长短期记忆网络（BiLSTM）单元提取胶囊网络的基本特征，并使用 k-means 算法对向量特征进行聚类以实现路由函数。实验表明，该模型不仅提高了 DGA 域名识别能力和 DGA 域名家族识别能力，而且在模型测试中展现出出色的实时发现能力。Zao[19]等人提出了一种域语言语音检测（DOLPHIN）方法。他们充分考虑到检测的上下文以及单词的发音和拼写之间的对应关系，设计了海豚模式。与现有的基于语言特征的方法相比，DOLPHIN 可以配合大多数语言特征，在性能上有很大的提高。但是对于某些 DGA 域名家族检测效果差。

总结来看，机器学习的传统方法在应用上存在操作繁复之处，如需人工分

析域名特征，使得识别过程耗时较长，进而影响了技术的进步。相对而言，深度学习的技术框架则能够有效地解决这些问题，优化了检测流程的效率。尽管如此，深度学习在识别过程中对特征的提取还是显得较为简单化，对于以单词表为基础生成的 DGA 域名的检测效果不佳。在检测小规模 DGA 家族和新兴 DGA 变体方面表现不佳，需要足够的训练数据支持[20]。主要可以总结为以下 3 种：

（1）部分 DGA 家族域名样本稀缺

DGA 家族之间存在严重的数据不平衡。大型家族可以达到数十万户，而小规模家族只有 100 户甚至更少[21]。对于小规模的 DGA 家族和新兴的 DGA 变体来说，获取足够的样本既费时又困难。传统的机器学习和深度学习通常在模型训练时利用已有的 DGA 域名数据。然而，新型 DGA 域名的出现导致相关样本匮乏，这使得模型在训练过程中性能多样性不足，这限制了其在准确性检测和最大化其深度学习潜力方面的效果。因此，对于小样本 DGA 域名进行数据增强以提高对新型 DGA 域名的检测能力十分重要。

（2）短 DGA 域名检测效果不佳

由于长度较短的域名具有较少的字符组合，从而导致特征信息较少，检测变得十分困难。同时，Fu[22]等人提出了两种较难检测的 DGA 域名，称为 SDGA 域名。这种域名的长度很短，特征提取困难，难以提取足够差异的特征。此外，SDGA 域名的数量很少，样本的不平衡性也增加了分类的难度。

（3）基于单词表的 DGA 域名检测效果不佳

基于单词表的 DGA 域名是从频繁变化的字典中选择单词来生成类似于正常域名的域名，与正常域名非常相似，难以识别。在自然语言处理中，单词通常会经过预训练生成词向量。然而，DGA 域名分析通常依赖于字符级别的嵌入，而不是基于预训练的单词级嵌入。这种方法限制了域名中单词的语义表达，导致了针对单词列表生成的 DGA 域名识别效率不高。

1.3 论文主要内容

综上所述，本文主要针对上面提出的三种 DGA 域名检测的不足，提出自己的解决方案。主要研究内容如下：

（1）针对部分 DGA 域名家族样本稀缺的问题，设计了一种基于生成对抗网络的 DGA 域名生成算法。该算法采用了 n-gram 字典和 WGAN 相结合的方

法，成功地生成了与目标域名高度相似的生成域名。通过将这些生成域名与现有数据集融合，显著提升了对 DGA 域名家族的识别效率，尤其在小样本数据集上，展示了其增强模型训练和提高识别精度的有效性。

（2）针对长度较短和基于单词表生成的 DGA 域名检测效果较差的问题，设计了一个基于域名长度的异构的 DGA 域名检测模型，根据域名长度分为字符级域名检测模块和单词级域名检测模块，该模型将不同的特征工程和深度神经网络集成在一个统一的检测方法中。对于字符级域名，采用了注意力递归图的方式进行数据处理，设计了并行多路卷积特征处理模块，实现了基于 DGA 的高精度域名识别，在短域名的检测上取得了显著的效果。对于单词级域名，采用了 n-gram 特征替代传统的手工提取特征方法，GRU-Attention 的特征提取，在检测效果上取得了更好的表现，从而提升了基于单词表的域名检测能力。

1.4 论文组织结构

论文后续内容分为三章，组织结构如图 1.1 所示。

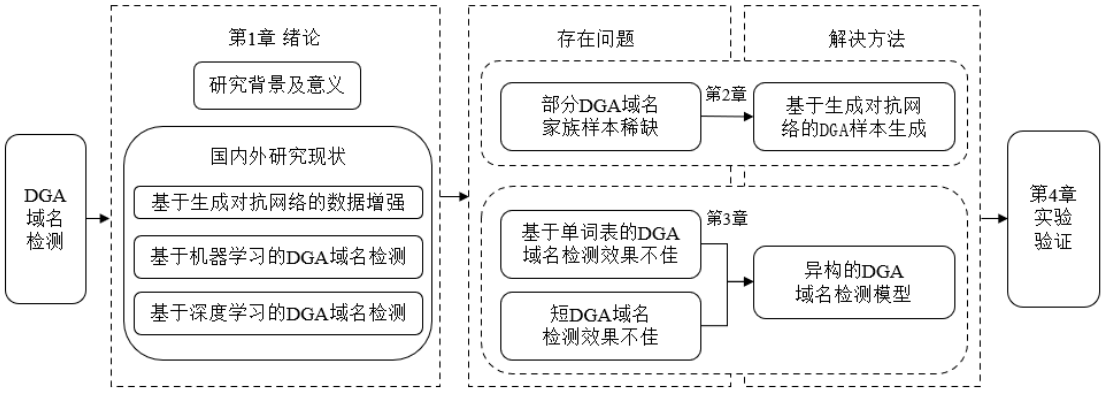


图 1.1 论文组织结构

具体分布如下：

第二章介绍一种基于生成对抗网络的 DGA 域名生成方法，详细介绍了模型的创新所在，生成了与真实域名高度相似的生成域名，可用于小规模 DGA 家族数据增强。

第三章介绍了一个基于域名长度的异构的 DGA 域名检测模型，针对字符级域名和单词级域名，使用不同的特征提取方法和检测方法，提升了对短域名和单词级域名的检测能力。

第四章对前两章提出的两种 DGA 域名模型进行了实验验证。结果表明，本文提出的方法能够更好地进行 DGA 域名检测和小样本数据增强。在对比其他现

有模型时，本文的检测方法显著减少了误识率，提高了准确率，凸显出本文所搭建模型的优异性能。

最后的总结部分总结了本文的工作，并分析讨论了本文工作的不足之处，提出了可能的技术路径和未来研究的展望。

第 2 章 基于生成对抗网络的 DGA 域名样本生成

2.1 问题的提出

在 DGA 域名检测研究中，训练集中某个 DGA 域名家族的样本数量较少时，会对检测分类模型的准确率产生负面影响。原因在于样本的稀缺限制了模型学习到的特征的多样性，进而影响了模型的泛化能力。特别是对于深度学习模型，尽管其高容量特性有助于捕捉复杂特征，但样本量的不足也可能导致过拟合问题。数据增强是应对样本不足的一种有效技术，它通过合成新样本来丰富训练集，可以提升模型的泛化性能并克服过拟合，特别是在不减少模型的表示能力的前提下。然而，传统的文本数据增强方法，比如随机替换和同义词替换，可能不适用于 DGA 域名数据。这是因为域名字符串有其独特的结构，包括长度和字符间的可读性，简单的文本替换操作可能会导致标签误标，从而降低训练数据质量。

鉴于此，本章节提出了一种基于生成对抗网络的 DGA 域名生成模型，专门针对小样本 DGA 域名数据进行数据扩充，以提高模型的检测率。该模型利用 n-gram 字典和 WGAN 结合的方法，确保生成模型能够稳定地收敛并生成与目标相似的域名。

本文提出的模型相较于传统的 GAN 模型的优势在于：使用的 n-gram 字典取自真实域名，增加了生成域名的真实性；在 GAN 架构基础上引入 BiLSTM，增加了对数据的特征提取效果，提升了获取远距离特征依赖关系的能力。这些改进提升了生成数据的真实性、准确性，为以后深入开展检测实验做铺垫。

2.2 域名分析与序列学习

本节探讨了利用 n-gram 分析域名数据和 BiLSTM 网络处理域名数据的方法，这些技术为理解 DGA 域名提供了有力的工具，为生成与真实域名高度相似的生成域名提供了基础。

2.2.1 n-gram 分析域名原理

n-gram[23]是一种应用广泛的基于统计语言模型的算法。此方法的核心在于通过设定的窗口尺寸滑动整个文本，以此创建一系列的定长片段。通过这种方式，n-gram 模型可以捕捉文本中的局部特征和词语之间的关联性。在 n-gram 模

型中，目标词的相关信息被聚集在 n 个词上，然后通过计算含有指定词的语句出现的概率来判断语句的合理性。通过这种方式，n-gram 模型可以用于语言建模、文本生成、词语预测等任务。

域名检测中运用 n-gram 算法以字符为最小单元，为减少数据的重复性，仅考虑二级域名。现在有 n 个字符组成的域名，如公式（2-1）所示：

$$p(s) = p(w_1, w_2, w_3, \ldots, w_n) \tag{2-1}$$

其中，s 是这 n 个字符组成的域名。给定词汇位置的不确定性导致必须依赖条件概率及其链式法则来估计语句的概率值：

$$p(w_1, w_2, w_3, \ldots, w_n) = p(w_1)\, p(w_2 |\ w_1) \cdots p(w_n |\ w_1, \cdots, w_{n-1}) \tag{2-2}$$

当序列非常长的时候，计算的复杂度会呈指数级别增长。而且数据稀疏十分严重，没有足够大的预测能够满足多个字符的共现，即不满足大数定律而导致概率失真。于是便引入了马尔可夫假设：一个字符的出现只与前面 n-1 个字符相关：

$$p(w_1 \cdots w_n) = \prod p(w_i |\ w_{i-1} \cdots w_1) \approx \prod p(w_i |\ w_{i-1} \cdots w_{i-N+1}) \tag{2-3}$$

如果一个字符的出现仅依赖于它前面出现的一个字符，就是一阶马尔科夫链（Bigram）：

$$p(S) = p(w_1 w_2 \cdots w_n) = p(w_1)\, p(w_2 |\ w_1) \cdots p(w_n |\ w_{n-1}) \tag{2-4}$$

如果一个字符的出现仅依赖于它前面出现的两个字符，那么就是二阶马尔科夫链（Trigram）：

$$p(S) = p(w_1 w_2 \cdots w_n) = p(w_1)\, p(w_2 |\ w_1) \cdots p(w_n |\ w_{n-1} w_{n-2}) \tag{2-5}$$

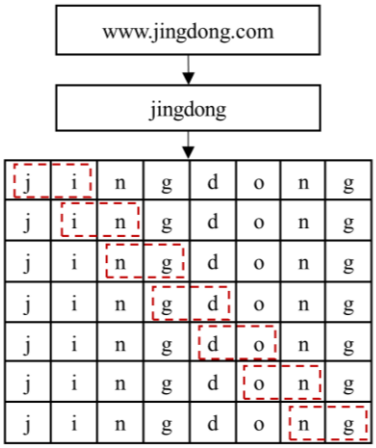


图 2.1 域名处理过程

综上，在分析具体字符串时，通过应用 n-gram 技术，可以得到由该字符串派生的所有 n 个字符长度的子串集合。以“www.jingdong.com”为例，设置 n 为 2，其分解的详细步骤在图 2.1 中展示。这项技术可广泛用于分析域名中。

2.2.2 BiLSTM 处理域名原理

单个 RNN 层以多个时间步展开，可以将其视为神经网络在时序上的权重共享，类似于 CNN 在空间上的权重共享。LSTM 也适用相同的原理。

在反向传播过程中，RNN 存在梯度消失问题。这是因为 RNN 在时序上共享参数，导致梯度在反向传播过程中不断相乘，从而导致梯度要么越来越小，要么越来越大。如果梯度值变得非常小，网络层将停止学习，尤其是较早的层。这些层不再学习，导致 RNN 无法在较长序列中保持信息，因此其记忆主要是短期的。RNN 时间线展开图如 2.2 所示。

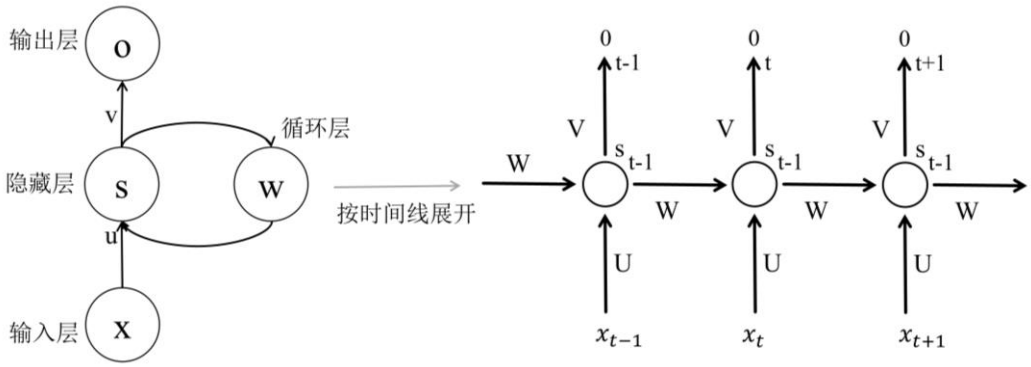


图 2.2 RNN 时间线展开图

LSTM 是由 RNN 网络发展而来的，具有比 RNN 更进一步学习依赖关系的能力[24]。LSTM 最早由 Hochreiter[25]等人于 1996 年提出，之后不断得到改进。该算法在解决各种问题时表现出色，尤其在处理字符串问题方面效果显著。

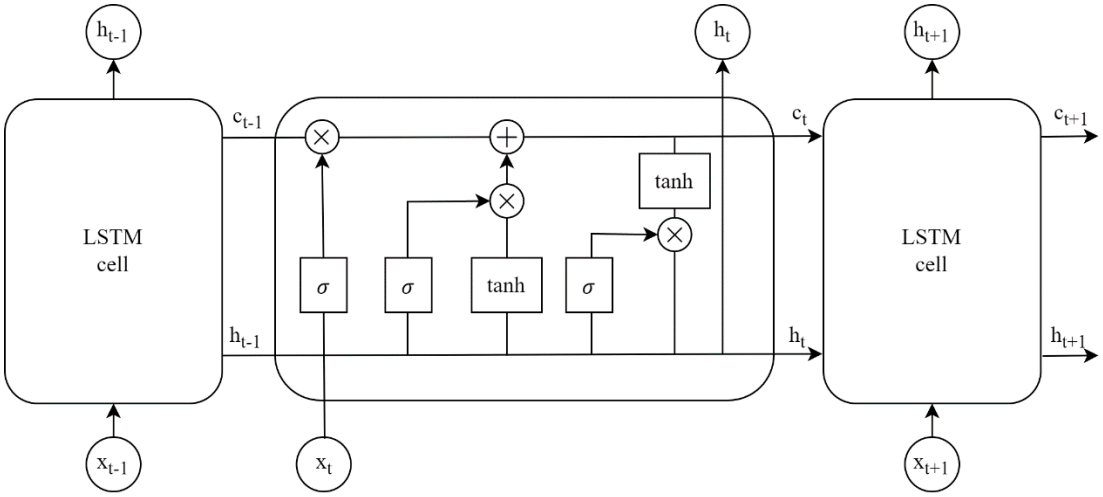


图 2.3 LSTM 结构图

LSTM 的结构也是基于链结构，但链中的每个模块都经过了更多的改进。与 RNN 不同，LSTM 在每个模块中有四个神经网络层，并且这些层之间可以相互作用。LSTM 的四层示意图如图 2.3 所示。

- 箭头：表示矢量在模型中的方向；
- 矩形：代表一个神经网络层，激活函数可以是 Sigmoid[26]或 tanh 层；
- 圆圈：表示矢量之间的计算，通常是矢量加法；
- 连接线：表示合并，复制线表示其数据被复制并移动到不同的位置。

LSTM 是为了克服短期记忆问题而提出的解决方案。通过“门”这种机制，LSTM 可以在长序列中传递相关信息，从而执行准确的预测。但是 LSTM 无法处理从句末到句首的逆向信息流。BiLSTM[27]通过合并序列的向后和向前上下文信息解决了这个问题。因此，采用 BiLSTM 网络对域名中隐藏的语义关系进行建模。BiLSTM 同时从前面和后面的字符子序列中获取信息。BiLSTM 网络首先用真实域名进行训练，在给定的特定前后字符子序列对下，学习每个可选字符的条件概率，从而形成域名。然后，在给定一对字符子序列后，BiLSTM 网络可以计算出每个可选字符的条件概率。

对于一般的任务，BiLSTM 网络的输入是一个序列。BiLSTM 通过一定数量的 BiLSTM 层学习输入序列的前向和后向。如图 2.4 所示，每个 BiLSTM 层由 LSTM 网络组成。每个 BiLSTM 层的输出将从层的输入的前向和后向学习到的输出连接起来。当前 BiLSTM 层的输出作为后续 BiLSTM 层的输入，而当前 BiLSTM 层的输入来自之前 BiLSTM 层的输出。最后一个 BiLSTM 层提供一个输出。因此，最终的输出是由从初始输入序列的前后方向学习到的信息得到的。

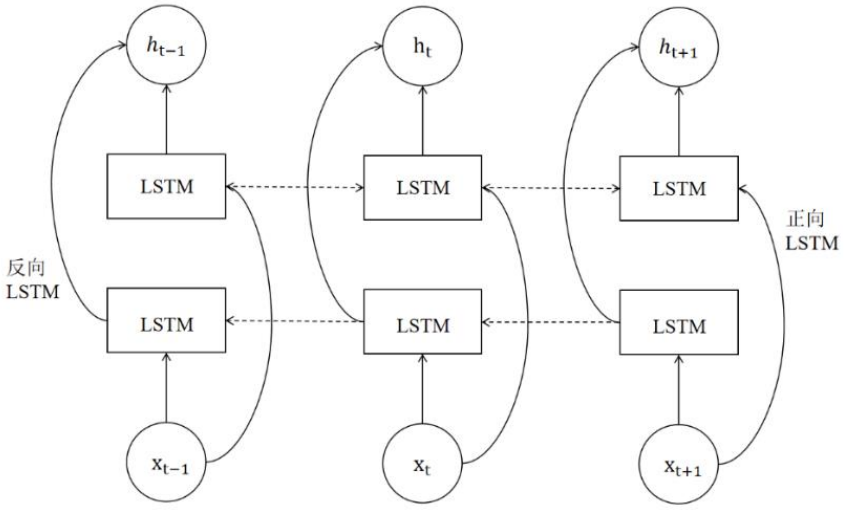


图 2.4 BiLSTM 结构图

因此，BiLSTM 能够有效处理序列数据，尤其适用于识别和学习 DGA 域名中的时间和顺序依赖性。通过前向和后向处理域名序列，BiLSTM 网络能够从两个方向捕获上下文信息，从而更全面地理解域名的结构和生成逻辑。这对于了解 DGA 生成的域名至关重要，了解 DGA 域名后才能生成与真实域名高度相似的生成域名。

2.3 基于 GAN 的 DGA 域名生成模型

n-gram 模型利用已存在的单词或字符序列来估计随后单词或字符的发生概率，能够捕捉序列中的局部特征和前后关系。在之前的研究中，Yang[28]提出了一种名为 N-Trans 的新型并行检测模型，该模型基于 N-gram 算法和 Transformer 模型，能够有效提取字母组合特征，捕捉字母在域名中的位置特征。在实验中，它可以有效、准确地识别恶意域名，并优于主流的恶意域名检测算法。Yun[29]等人提出了一种基于神经语言模型和 WGAN 的新型 DGA，也产生了较好的效果。因此受这些人的启发提出了一个基于 n-gram 字典和 WGAN 的 DGA 样本生成模型 NWGAN。

2.3.1 模型概述

NWGAN 的总体思想是选择在真实域名中出现概率高的 n 个 gram 作为基本单位，根据 n 个 gram 在真实域名中的排列自适应合成新域名，为此，NWGAN 包含三个模块来实现其功能：

n-gram 生成器：首先从真实域名中提取最常见的 n-grams，并将这些 n-grams 构建成一个字典，这些 n-grams 是合成新域名的基本单元。

域名处理器：使用上述字典将真实域名分割成 n-grams，为下一步的域名合成做准备。为此，在域名处理器中应用了一种称为双向最大匹配方法的标记化算法，将真实域名字符串的基本单位从字符变为 n-gram，这是恢复真实域名中 n-gram 之间相关性的准备工作。

域名生成器：核心模块，基于 WGAN 原理，学习真实域名中 n-grams 的排列规律，进而合成类似于真实域名的新域名。包括一个生成器和一个判别器，两者在训练阶段通过相互竞争来优化自己的目标函数，最终得到一个能够生成与真实域名相似的生成器。过程如图 2.5 所示。

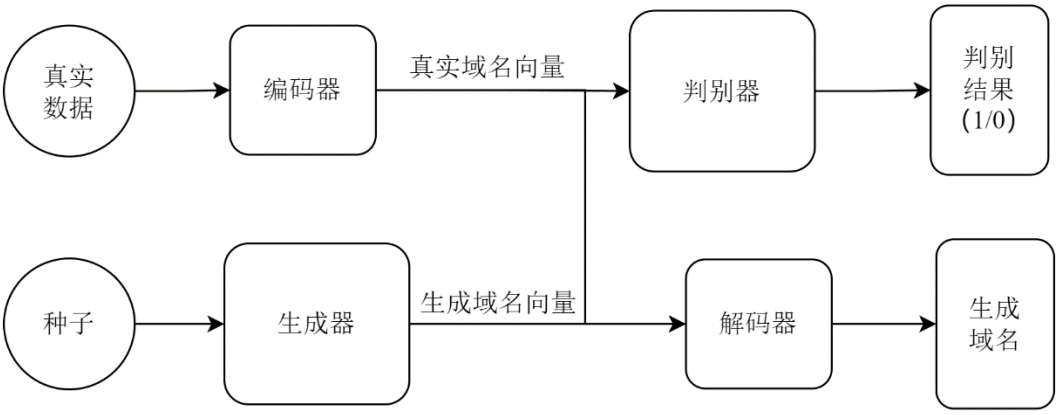


图 2.5 域名生成器

2.3.2 n-gram 生成器

n-gram 生成器从真实域名列表中提取 n-gram，并构建包含这些 n-grams 的字典。这些 n-grams 被用作合成新域名的基本单位。当输入真实的域名到这个模块，它为 n-gram 生成器模块输出一个 n-gram 字典。

首先，从真实域名中提取 n=1、2、3、4 的 n-grams，接下来，计算每个 n-gram 的出现次数，并按次数升序对这些 n-gram 进行排序。最后，将前 5000 个 n-gram 收集到一个字典中。通过这种方式，得到了一本字典，其中包括最常见的音节和首字母缩略词在真实的域名，而不需要任何语言的知识。对于域名，例如“mail.google.com”，将“.com”称为顶级域名（Top-Level Domain，TLD），将“google.com”称为二级域名（Second-level domain，SLD），将“mail.google.com”称为三级域名（Third-level domain，3LD）。其中，“google”为二级域标签，“mail”为三级域标签。DGA 不需要创建 TLD，因为所有合法域名的 TLD 都是从根区域数据库中采样的。为方便起见，在本工作中将重点放在二级域标签上，称其为“域名”。

2.3.3 域名处理器

域名处理器的作用是根据 n-gram 生成器生成的字典，将真实域名分割成一个 n-gram 数组。这个模块接收到的输入是送往 n-gram 生成器的真实域名集合，输出则是经过标记化处理的真实域名列表。

具体来说，域名处理器使用的是双向最大匹配法来对域名进行分割，这是一种常用于中文句子分词的方法。由于域名和中文句子都不使用空格来分隔基本单元，因此认为双向最大匹配法也适用于域名的分割。该算法结合了两种分

割方法：正向最大匹配法和反向最大匹配法。前者从左至右尽可能长地将域名分割成字典中的 n-grams，后者则从右至左进行。双向最大匹配法通过以下启发式规则从两种方法中选择更好的结果：

（1） 如果两种结果中 n-grams 的数量不相等，选择 n-grams 较少的那个结果。

（2） 如果两种结果中 n-grams 的数量相等，

① 如果两个结果相同，则随机选择一个；

② 如果两个结果不同，则选择个别字符较少的那个结果。

由于字典中包含了所有在域名中出现的字符，所以在这一过程中不会出现字典外的 n-gram。

2.3.4 域名生成器

（1）编码器和解码器

由于神经网络的设计无法直接处理字符串格式的数据，因此需要利用编码器将域名转化为张量，即域嵌入（domain embeddings）。具体方法是对字典内每个 n-gram 实施 one-hot 编码，并将这些编码按序排列来表示域名，形成一个二维张量。考虑到实际域名的长度差异，引入了代表空白的特殊编码，在短于最大长度的域名尾部进行填充，以达到长度一致性。

在生成器的输出端设置了一个解码器，负责将域嵌入逆转回域名字符串。该解码器运用 argmax 函数确定每个位置嵌入值最大的索引，据此找到相应的 n-gram，最后将这些 n-gram 拼接形成完整的域名。

argmax 是一个数学和计算机科学中常用的函数，用于找到函数的输入值，使得函数的输出值达到最大值，arg 即 argument，此处意为“自变量”。当有个函数 $y=f(x)$ 时，$\operatorname{argmax}(f(x))$ 是使得 $f(x)$ 最大的那个变量 x；若有多个点使得 $f(x)$ 取得相同的最大值，那么 $\operatorname{argmax}(f(x))$ 的结果就是一个点集。换句话说，$\operatorname{argmax}(f(x))$ 是使得 $f(x)$ 取得最大值所对应的 x 的集合。

argmax 的公式如下：

$$\underset{x\in S\subseteq X}{\operatorname{argmax}} f(x) \coloneqq \{x|\ x\in S\wedge \forall y\in S: f(y)\le f(x)\} \tag{2-6}$$

对一个函数 $f(x)$ 或一个映射 $f: X\to Y$，当 x 取值范围为 S 的时候（也叫 $x\in S$），argmax 的结果是使得 $f(x)$ 取得最大值的 x 点集。所以如果明确指出 $x\in S$ 的话，则表示并非在所有 $f(x)$ 的输入变量范围内进行最大结果值搜索。

当 $S=X$ 或者根据上下文 S 已知的时候，可以将公式简化成：

$$\operatorname{argmax} f(x) := \{x \mid \forall y : f(y) \le f(x)\} \tag{2-7}$$

举例：若有函数 $f(x) = 1 - |x|$，则 $\operatorname{argmax} f(x)$ 的结果为{0}。对比 $\max f(x)$ 函数，其定义为：

$$\max_x f(x) = \{f(x) \mid \forall y : f(y) \le f(x)\} \tag{2-8}$$

（2）生成器和鉴别器

在生成对抗网络中，有两个核心组件：生成器 G 和鉴别器 D。它们共同协作来实现对抗训练，从而生成具有高质量的数据样本。生成器负责将低维度的随机向量映射为高维度的目标数据样本，生成所需的域名。鉴别器的任务是将生成器生成的域名与真实域名进行区分。通过这种零和博弈的方式，鉴别器不断提高其鉴别能力，同时生成器也在努力提高生成的样本的质量。本节详细介绍了生成器和鉴别器，具体结构如图 2.6 所示。

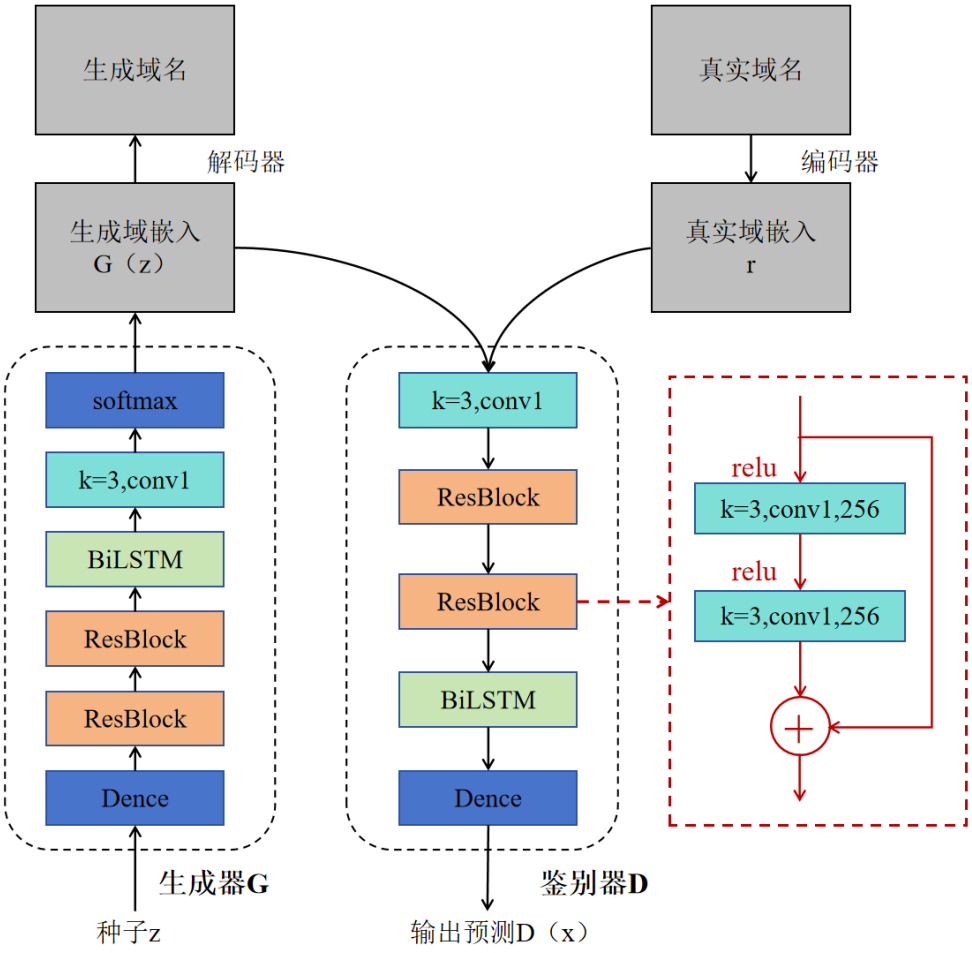


图 2.6 生成器和鉴别器

以生成器为例，生成器的输出是一批域嵌入，这些域名嵌入是通过将随机

采样的种子 seed 经过网络生成的。具体而言，生成器接收的输入是一个随机值填充的张量，这个张量代表了生成过程的种子，它被映射到一个高维空间，以此作为生成域嵌入的起点。如生成器接受种子 z 作为输入，全连接层将输入的种子向量映射到更高维度的表示空间，为后续的卷积神经网络提供初始特征表示。卷积神经网络包含了 2 个残差块，每个残差块由 2 个卷积模块组成，共计 4 个卷积模块。每个卷积模块都包含卷积层、批归一化层、Dropout 层和 ReLU 激活函数。这些层的结合能够有效地学习数据的特征，并减少训练过程中的梯度消失和过拟合问题。

尽管卷积神经网络在捕获局部特征方面表现出色，但其在捕获远距离依赖关系方面存在一定的局限性，特别是当距离较远时。在卷积神经网络层之后加入 BiLSTM 是一种有效的增强模型对于长距离依赖关系建模能力的策略。与传统的单向 LSTM 相比，BiLSTM 能够捕获到序列中每个字符的双向上下文特征，从而更好地建模序列数据中的复杂依赖关系，使生成器能够更好地生成符合预期的序列数据。

在双向体系结构中，有来自两个独立 LSTM 的两层隐藏节点，两个 LSTM 从不同的方向捕获依赖关系。第一个隐藏层有来自最后一个单词的循环连接，而第二个隐藏层的循环连接方向是颠倒的，在序列中向后传递激活。因此，在 LSTM 层，可以从前向 LSTM 网络中获得前向隐藏状态，也可以从后向 LSTM 网络中获得后向隐藏状态。双状态从字符序列的两个方向捕获组合语义信息。LSTM 存储单元的实现如式（2-9）所示：

$$\begin{aligned} f_t &= \sigma\left(W_f \cdot \left[h_{t-1}, m_t\right] + b_f\right) \\ i_t &= \sigma\left(W_i \cdot \left[h_{t-1}, m_t\right] + b_i\right) \\ \tilde{C}_t &= \tanh\left(W_i \cdot \left[h_{t-1}, m_t\right] + b_i\right) \\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\ o_t &= \sigma\left(W_o \cdot \left[h_{t-1}, m_t\right] + b_o\right) \\ h_t &= o_t * \tanh\left(C_t\right) \end{aligned} \tag{2-9}$$

在公式（2-9）中，符号 f 、i 、o 和 C 依次代表遗忘门、输入门、输出门以及细胞状态向量。相应的，特定的权重矩阵用于链接这些门到隐藏层，确保数据流的正确处理。σ 指的是逻辑 Sigmoid 激活函数。

BiLSTM 由正向 LSTM 和反向 LSTM 组成。CNN 层的输出向量为 $c = \{c_1, c_2, c_3, ...c_n\}$，正向 LSTM 读取输入向量从 $\{c_1\}$ 到 $\{c_n\}$，反向 LSTM 读取输

入向量从$\{c_n\}$到$\{c_1\}$，即反向顺序，同时生成一对隐藏状态$\left(\vec{h}_1,\vec{h}_2,\ldots\vec{h}_z\right)$和$\left(\overleftarrow{h}_1,\overleftarrow{h}_2,\ldots\overleftarrow{h}_z\right)$。将两个隐藏状态进行组合得到 BiLSTM 层的输出，如（2-10）所示。

$$h_i=\left[\vec{h}_i,\overleftarrow{h}_i\right]^T \tag{2-10}$$

生成器的输出首先是一维卷积，一维卷积层的作用是将两个残差块的输出映射到域嵌入空间，每个 n-gram 编码为特定 n-gram 的 one-hot 编码。最后，在卷积层之后应用一个 Softmax 层。这意味着 Softmax 层将卷积网络的输出转换为一个概率分布，其中每个输出位置的概率分布对应于字典中所有可能 n-gram 的概率。通过选择每个位置概率最高的 n-gram，模型能够为合成的域名生成一系列特定的n-gram，这些n-gram随后被解码器连接成完整的域名。因此，Softmax 层是将生成模型的连续输出转换为离散的 n-gram 选择的关键组件，确保了生成的域名由学习到的 n-gram 模式构成，从而增加了生成域名的多样性和真实性。在生成器的输出$G(z)$处设置一个解码器，将域嵌入转换为域名。解码器通过使用 argmax 函数获得嵌入值最大的索引，找到与索引对应的 n-gram，并将 n-gram 连接成域名。

鉴别器的输入是来自两个来源的域名：一部分是真实的域名，另一部分是生成器生成的域名。为了处理这些域名，鉴别器首先需要将它们转换为域嵌入，然后这些域嵌入被输入到鉴别器的神经网络中，网络通过一系列卷积层来学习和提取域名嵌入中的局部模式。鉴别器与生成器具有相似的网络构架，包括卷积层及 BiLSTM 层。在训练期间，为了初始化模型参数，鉴别器与生成器将首先进行预训练。输出部分结束于一个全连接节点，以完善模型的输出接口。

（3）生成网络 WGAN

生成器 G 的目标是将从先验分布P_z中随机采样的种子 z 转换成一个生成样本$G(z)$，使其看起来像是来自真实数据分布P_r。鉴别器D的目标是正确判断样本x是否来自真实数据分布P_r，或者是由生成器 G 合成的，并输出一个概率$D(x)$，表示样本x来自真实数据分布P_r的概率。

GAN 为这两个模型设置了一个零和博弈。训练鉴别器 D 使标记样本的真阳性率和假阳性率之和最大化，训练生成器 G 使假阳性率最大化。将此博弈表示值函数$V(D,G)$的极大极小问题：

$$\min_G\max_D V(D,G)=E_{x\sim P_r}[\log(D(x))]+E_{z\sim P_z}[\log(1-D(G(z)))] \tag{2-11}$$

当鉴别器为最优时，公式(2-11)可以重新表述为最小化生成数据分布 P_g 与真实数据分布 P_r 之间的 JS 散度的问题。

鉴别器 D 的训练过程如下：

① 首先输入一个从先验分布 P_z 随机抽样的张量 z 作为种子到发生器 G，发生器 G 产生一个合成样本 G(z)。

② 将真实数据分布 P_r 中的合成样本 G(z)和真实样本 r 输入鉴别器 D。

③ 冻结发生器的参数，并按照公式上升其随机梯度来更新 D 的参数。

④ 重复步骤①到③，直到鉴别器 D 的参数收敛。

生成器 G 的训练过程如下：

① 输入一个从先验分布 P_z 随机抽样的张量 z 作为种子到生成器 G。

② 冻结 D 的参数，并通过公式中的梯度下降来更新 G 的参数。

首先训练鉴别器，然后训练生成器。重复这两个过程，直到 GAN 中的所有参数收敛。

不幸的是，当用 JS 散度作为衡量分布的距离的工具时，存在明显的局限性，在某些情况下，P_q 和 P_r 之间的 JS 散度不能为训练神经网络提供任何有用的梯度方向，例如当它不连续时。可能导致生成模型偏向于生产较为保守的输出，以符合要求，进而降低了域名的多样性。这些挑战使得传统的 GAN 训练过程显得尤为艰难。为了解决这个问题，Arjovsky[30]等人提出了 WGAN。

WGAN 的优点在于它的训练过程定义明确，使得生成器易于训练。WGAN 用 Wasserstein 距离来描述 P_g 和 P_r 之间的距离：

$$W\left(P_g,P_r\right)=\inf_{\gamma\sim\Pi\left(P_g,P_r\right)}\mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|] \tag{2-12}$$

在公式（2-12）中，Wasserstein 距离 $W\left(P_g,P_r\right)$ 定义为生成数据分布 P_g 和真实数据分布 P_r 之间的最小成本，它涉及所有可能的联合分布 $\Pi\left(P_g,P_r\right)$ 的边缘分布分别是 P_g 和 P_r。由于 Wasserstein 距离在这种情况下是连续和可导的，有效解决了 GAN 训练过程中的梯度消失问题，使模型训练更加稳定。

由于难以精确地求出最小值，将带入对偶问题：

$$W\left(P_g,P_r\right)=\frac{1}{K}\sup_{\|D\|_L\le K}\left(\mathbb{E}_{x\sim P_r}[D(x)]-\mathbb{E}_{x\sim P_g}[D(x)]\right) \tag{2-13}$$

其中，由鉴别器表示的函数 D 必须满足 1-Lipschitz 连续性，即存在一个正实常数 K，使得对于所有实数 x1 和 x2，$|D(x1)-D(x2)|\le K|x1-x2|$。本文用的此种方法。

2.3.5 用 NWGAN 生成数据

在使用 NWGAN 时，向生成器输入一个种子。生成器使用种子自动生成张量，然后根据 n-gram 字典将张量解码为域名。

首先通过离线训练提前获得生成器和 n-gram 字典。在训练过程中，向 n-gram 生成器输入一个真实域名列表（如“google”、“baidu”、“taobao”）。n-gram 生成器在实际域名中选择出现频率最高的 5000 个 n-gram，然后使用这些 n-gram 作为 n-gram 字典的构建块。字典中包含 n 个 n∈{1,2,3,4}的 grams，如“g”、“oo”，“goo”，“gle”。根据字典，n-gram 标记器将每个域名标记为 n-gram 序列，例如，“google→goo/gle”，“baidu→ba/idu”，“taobao→tao/bao”。域嵌入将 n-gram 字典中的每个 n-gram 编码为一个 one-hot 编码（长度等于字典大小的向量），然后通过将域名中 n-gram 对应的 one-hot 编码连接起来，将每个标记化的域名转换为矩阵。如图 2.7 所示。

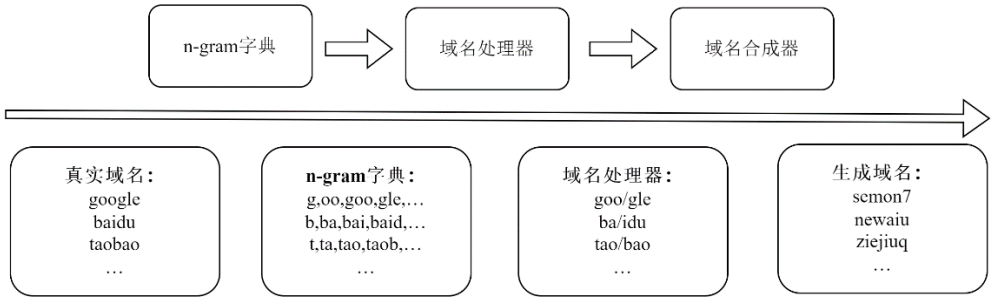


图 2.7 域名生成过程

具体过程如下：

① 建立 n-gram 字典：从真实域名中提取并选择最常见的 n-gram，建立一个包含这些 n-gram 的字典。在 NWGAN 模型中，选择了最常见的 5000 个 n-gram 来构建这个字典。

② one-hot 编码：对于字典中的每个 n-gram，创建一个 one-hot 编码，这个编码的长度等于字典的大小（即 5000）。如果 n-gram 是字典中的第 i 个元素，那么在其对应的 one-hot 编码中，第 i 个位置为 1，其他位置为 0。

③ 转换标记化域名为矩阵：对于一个标记化的域名，将其分割成的 n-gram 转换为相应的 one-hot 编码，并将这些编码连接起来形成一个矩阵。这个矩阵的每一行代表一个 n-gram 的 one-hot 编码，整个矩阵代表了整个域名。

通过这种方式，NWGAN 能够将域名转换为可以被神经网络处理的数值型数据（矩阵），为后续的生成过程提供基础。这个矩阵随后被用作生成新域名

的基础，其中生成器试图模仿这个矩阵的结构来生成新的、看起来像真实域名的矩阵，这些矩阵最终被转换回文本形式的域名。

然后，域合成器使用 WGAN 框架对表示标记化域名的矩阵进行训练，以生成与真实域名相似的域名。经过训练过程后，攻击者可以使用训练有素的生成器生成一批和真实域名极为相似的域名，如 semon7、newaiu 和 ziejiuq。达到扩充数据集的目的。

2.4 本章小结

本章针对部分 DGA 家族域名样本稀缺的问题，提出了一种基于生成对抗网络的域名生成算法，即 n-gram 字典与 WGAN 相结合的域名生成算法 NWGAN。同时，详细讨论了其包含的三个模块，并介绍了如何使用 NWGAN 生成数据。此外，利用 n-gram 作为域名核心特征提取技术，增强了对数据的特征提取效果，并引入了 BiLSTM 来捕捉 DGA 域名序列中的局部模式和上下文关系。这不仅增强了模型对 DGA 家族特有统计特征的识别能力，还能够产生与目标 DGA 家族域名高度相似的样本。

第 3 章 异构的 DGA 域名检测模型

前一章提出了基于生成对抗网络构建 n-gram 字典与 WGAN 相结合的域名生成算法，生成的 DGA 域名样本与真实恶意域名在特征层面表现出高度相似性，它们为检测模型训练带来了宝贵的数据资源。接下来，本章将介绍一种新型的异构 DGA 域名检测模型，并展示如何使用这些生成的样本与真实数据共同进行模型训练。

3.1 问题的提出

尽管基于深度学习的方法在识别恶意域名，尤其是由 DGA 生成的域名方面取得了显著成绩，但对于一些特定类型的 DGA 域名，例如短域名和基于单词表的域名，这些方法的检测效果并不理想。

短域名的检测。长度是决定域名是正常的还是基于 DGA 的一个极其重要的因素[31]。Ahluwalia[32]研究了长度对 DGA 域名检测的影响。域名长度的减少将导致检测模型的性能严重下降。Schüppen[33]等人提出 FANCI 是一种基于 DGA 域名和合法域名之间的字母数字字符分布差异的域名检测模型。图 3.2 显示了 FANCI 对不同长度的域名的检测结果。Avg-Acc（黑色）表示 FANCI 检测结果的平均准确率。PMF（棕色）是概率质量函数，表示具有不同长度的样本比例。CDF（红色）是累积分布函数，描述了具有不同长度的样本数量的分布情况。图 3.1 的横坐标表示子域的长度。

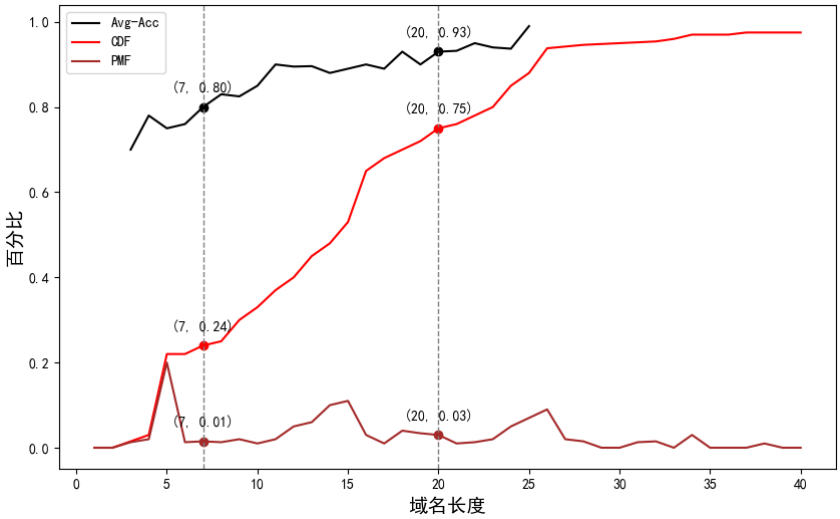


图 3.1 FANCI 检测基于 DGA 的不同长度域名的性能

由图 3.1 得到，域名长度在 1 至 7 个字符之间的 DGA 样本构成了总数的 24%。在这一特定长度范围内，FANCI 模型的检测效果并不理想，其准确率未能达到 80%。总体而言，FANCI 的检测性能随着域名长度的减小而下降。也就是说，FANCI 模型对样本长度很敏感。随后，Liang[34]等人通过分析不同长度样本的最新模型的性能趋势，发现模型对长度敏感。域名长度的减少将导致检测模型的性能严重下降。

基于单词表的域名检测。基于单词表的 DGA 域名通常通过组合意义完整的单词生成，使得它们在表面上更接近合法域名，从而难以被基于传统特征的检测模型识别。一些研究尝试通过融合不同的特征提取方法和检测模型来提高对这类域名的检测效果。

为了进一步提高对不同类型 DGA 域名的检测效果，引入了异构网络模型的概念。异构模型的概念是指利用多种不同类型的检测方法、技术或数据源来增强检测的效果。这种方法通过结合多个不同的检测机制，可以提高检测系统的鲁棒性和准确性，从而更好地应对不同类型和变种的攻击。Liang 等人提出了名为 HAGDetector 的异构 DGA 检测模型，该模型通过三种针对不同域名长度的特征提取技术，并构建了多样化的检测架构来优化这些技术的应用。另外，Wang[35]等人提出的 HANDOM 模型，使用了异构注意力网络（HAN）增强了信息处理的效率，融合统计数据特征与图形结构信息来弥补传统方法的不足，以实现高性能的恶意域名分类。

综上所述，通过整合多样化的特征和模型，可以显著提升对复杂 DGA 域名的检测能力，尤其是在处理短域名和基于单词表的域名方面。因此，本章提出了异构的 DGA 域名检测模型，根据域名长短并行的处理不同的数据类型和捕获复杂的关系，从而提高预测的准确性。

3.2 域名检测技术

3.2.1 卷积神经网络与域名数据分析

在深度学习中，CNN 因其独特设计而广泛应用于多种任务。CNN 通过局部连接和权重共享显著降低模型参数数量。与传统神经网络如多层感知器不同，CNN 专为处理数组形式数据设计，通过卷积操作从输入数据局部区域提取特征，模仿人眼图像感知。CNN 可以学习并存储输入输出间的复杂关系，在其过滤器权重中反映数据的抽象信息。此外，CNN 通过设计精良的卷积过程，有效捕捉

数据的空间层次结构，为图像识别、视频处理及自然语言处理提供支持。

标准的 CNN 架构包括若干主要组成部分：输入层、卷积层 Conv、池化层 Pool、全连接层 FC 以及输出层[36]。CNN 结构如图 3.2 所示。训练中，通过反向传播算法[37]优化参数。

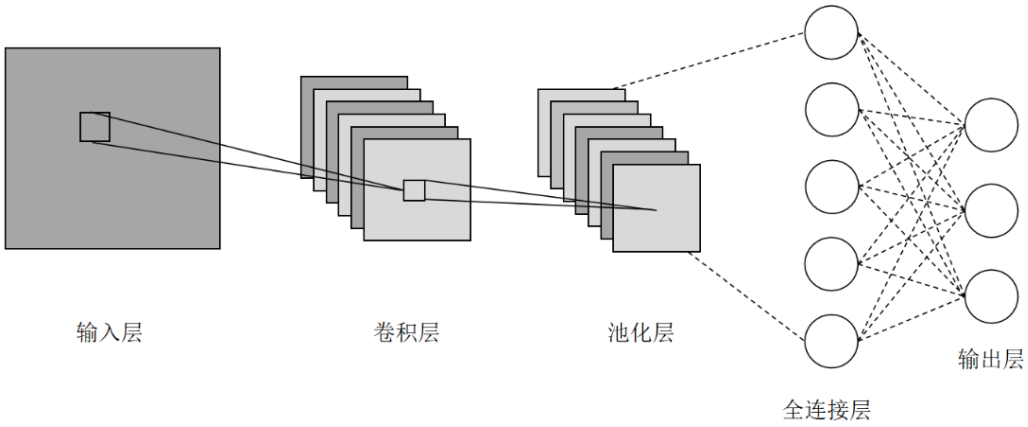


图 3.2 CNN 结构图

近年来，随着技术的不断发展，卷积神经网络在自然语言处理（Natural Language Processing，NLP）领域也逐渐展现出其强大的潜力，与 NLP 领域的契合度日益提升[38]。值得注意的是，在网络安全和自然语言处理领域中，应用 CNN 处理字符类型数据的方法十分相似。常规处理流程包括首先通过构建词向量捕获字符的内在和相互间的语义关系，继而通过 CNN 模型深入挖掘这些数据的语义特征[39]。最后，这些特征被送往全连接层进行精准的分类预测。所以，将 CNN 用在恶意域名检测方向是完全可行的。

3.2.2 递归图与域名动态分析

递归图（Recurrence Plots，RP）是一种可视化动态系统递归特性的工具。通过在相空间进行重构，递归图能够将相空间中的点转化为二维图形，直观地展现出来[40]。这种二维图形包含丰富的非线性信息。

相空间重构是构建递归图的核心步骤。在这一过程中，“相”指的是系统在特定时刻的状态，而所有可能状态的组合则构成了“相空间”。通过采用特定的方法和调整相关参数，相空间重构实质上是对系统时域信息的一种重新排列和变换，这一系列的操作能够将信号转化至更高的维度，并揭示出系统的某些特定特性。

相空间重构涉及到选择合适的嵌入维度 m、延迟系数 τ、和阈值 ϵ。这些参数的选择对于准确揭示动态系统的递归特性至关重要。选取嵌入维度 m 通常使

用伪邻域法，选取延迟系数τ常采用平均互信息法，至于最佳递归阈值ϵ的选择，目前没有一种普遍适用的方法。在实际应用中，通常根据递归图的峰值来选择阈值，一般选择峰值的10%作为阈值。这个阈值的选择会影响到递归图中点的连接情况，进而影响到对系统递归特性的分析。

在将递归图应用于时间序列时，首先需要将时间序列从时域空间变换到相空间，从而将时域中的每个点x_i映射成相空间的对应状态$\vec{s_i}$；接着计算每两个状态向量之间的距离（通常采用向量范数作为距离度量）；然后根据选定的阈值进行二值化处理，即如果两个状态向量之间的距离小于阈值，则在递归图中将这两个状态之间用线连接起来。这样，递归图就直观地展示了时间序列中状态之间的递归关系，从而揭示了系统的动态特性。递归图可以用一系列递归矩阵表示，如公式（3-1）所示：

$$R_{i,j}(\epsilon)=\Theta\left(\epsilon-\|\vec{s_i}-\vec{s_j}\|\right),i,j=1,\ldots,N \tag{3-1}$$

其中$R_{i,j}(\epsilon)$是递归矩阵的一个元素，表示时序数据中第i个和第j个点的关系。Θ是单位阶跃函数，用于判断两个状态点是否足够接近（即是否小于阈值ϵ），ϵ是一个预设的阈值，用来确定状态的相似性。$\|\vec{s_i}-\vec{s_j}\|$表示状态$\vec{s_i}$和$\vec{s_j}$之间的距离。$\vec{s_i}$和$\vec{s_j}$是相空间中的状态向量。

其算法流程如下：

① 由时间序列得到相空间状态集；

② 计算每两个状态之间的距离（向量范数）；

③ 进行阈值二值化，得到递归图矩阵。

递归图提供了一种直观表征序列图的方法，是一种强大的工具，用于揭示序列中的结构和模式，特别适用于那些具有周期性、重复性或复杂结构的数据。通过可视化和特征提取，研究人员可以更好地理解时间序列数据并进行进一步的分析。

3.2.3 注意力机制与域名特征处理

注意力机制作为神经网络领域的一项前沿技术，已在机器学习和自然语言处理任务中展现出其卓越的效能[41]。该技术可以敏锐地捕捉句子中单词间的关联性及其在句子结构中的核心作用，从而精确地定位句子的重点所在。通过全局视角对句子进行综合分析，并将单词与其上下文环境相协调，注意力机制实现了名为“自注意”（Self-Attention）的自我学习和优化过程。这种机制使模

型能够精准识别对输出标签具有显著影响的关键信息，进而集中资源对这些关键信息进行深度处理，从而有效提升预测的精确性。

注意力机制的核心组件包括查询（Query）、键（Key）和值（Value）三个要素。这些要素被精心设计，旨在协助模型在处理数据时更加聚焦于那些关键且重要的特征。通过引入注意力层，模型能够高效地筛选出对当前任务至关重要的信息，进而实现更为高效和精准的信息处理。注意力层的结构如图 3.3 所示。

查询 q 得到以下要处理的信息。它们分为两个部分，分别是键和值。这个信息用 k_i 和 v_i 表示。

对于每一个 q 的输入，a_i 称为第 i 个信息对 q 的影响，计算公式为：

$$a_i = \alpha(q, k_i) \tag{3-2}$$

然后将 a_i 归一化得到 b_i，常用的归一化函数是 Softmax。

$$b_i = \frac{\exp(a_i)}{\sum_j \exp(a_j)}, b = [b_1, b_2, \ldots, b_n]^T \tag{3-3}$$

最后，基于 b_i 重新计算 v_i 值。归一化可以防止数据与正在训练的模型相比被调整大小。注意机制有助于增加对数据本质特征的注意，这可以扩展到许多其他问题。

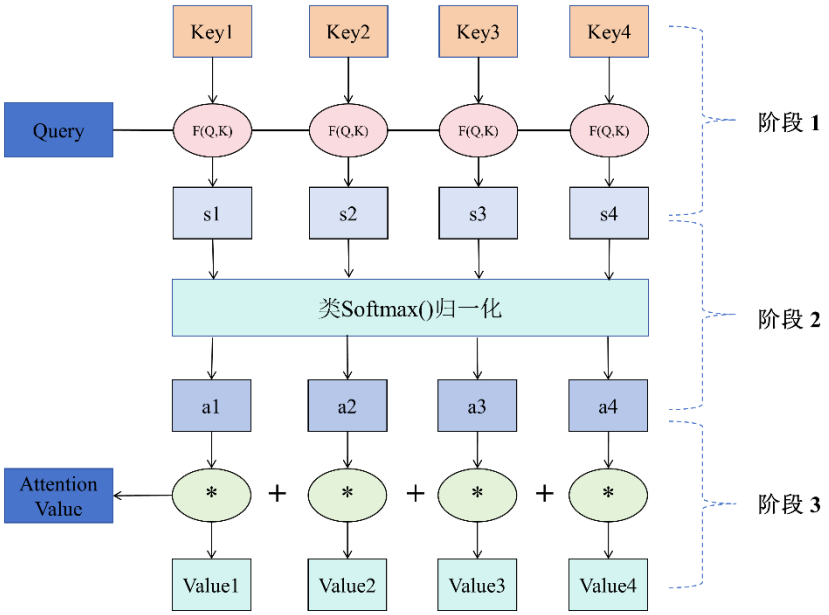


图 3.3 注意层的组成

注意力机制在自然语言处理领域的应用中展现出了显著的优点，这些优点同样适用于域名处理任务，具体使用如图 3.4 所示。因此，完全可以应用于

DGA 僵尸网络的检测问题。通过对 DGA 僵尸网络的分析，得出以下几点看法：

① 算法的输入是 DNS 查询或域名。如果将域名分解为单词或字母，则可以将域名视为一个句子。有些词可以代表主体，也就是说有些关键词可以代表一个域名的特征。

② 同一家族 DGA 僵尸网络中的域名具有相似的特征，建立在该域名家族的关键字集上。因此，不是相同家族的域族之间的具体关键字存在差异。

③ 同一家族的域名可以为该域名创建一个相似的环境，基于一组关键字的特征来生成该域名家族。在分类问题中，不同类之间的环境是不同的。

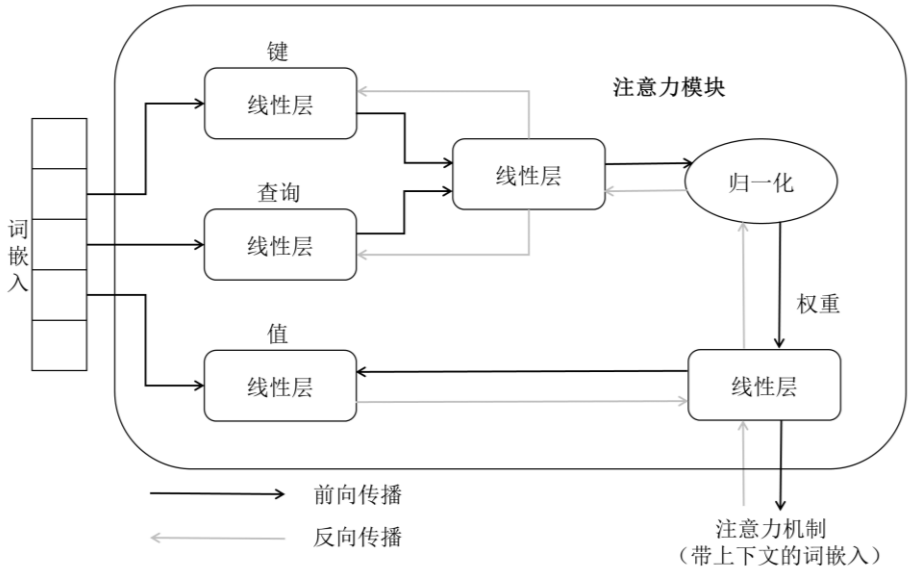


图 3.4 注意力在域名处理中的应用

3.3 异构 DGA 域名检测模型

为了有效提升短域名和基于单词表的 DGA 域名的检测与分类能力，本节引入了一种新颖的基于域名长度的异构 DGA 域名检测模型。这一模型整合了多样化的特征提取策略，融合了深度学习和自然语言处理技术的优势，从而开辟了一个全新的视角来辨识恶意域名。通过对模型结构的创新设计，不仅针对域名检测领域中的短板问题进行了专门的优化，还提供了一种更加灵活和全面的解决方案，显著提高了网络安全防御体系对复杂威胁形态的响应能力。

3.3.1 总体模型

异构 DGA 域名检测模型的设计起始于对域名长度的深入分析，根据域名的长度将其分为字符级域名和单词级域名两大类，并分别针对这两种类型采取了

定制化的特征提取策略。

针对短域名，模型采用了注意力递归图技术，将域名的特征映射到更高维度的空间中，以全面提取域名的细微特征。

基于单词表的域名长度一般比较长，n-gram 分析成为了一个强有力的工具，它能够有效识别域名中的模式，例如特定字母组合或单词片段，这在区分合法域名与 DGA 产生的域名时尤为重要。DGA 域名的随机性和不规则性在 n-gram 的帮助下得以揭露，从而使得模型能够识别出不符合自然语言习惯的字序列。模型的核心特性在于其集成了 GRU-Attention 机制的特征提取技术，这一复合机制将 GRU 对长期依赖关系的捕捉能力与注意力机制对关键信息的聚焦能力相结合，显著提升了模型对域名中重要部分的深入理解。这种方法的独特之处在于模型的主动性，它不再是被动地接受数据，而是能够积极识别并赋予数据中的关键信息以更高的权重。通过这种方式，模型能够实现对 DGA 域名的高精度识别，从而大大提高了域名检测与分类的准确性和效率。

整体而言，异构 DGA 域名检测模型通过多维度的特征分析，为 DGA 域名检测提供了一种有效的工具，它不仅能够处理复杂的数据结构，还能够应对日益精细化的僵尸网络攻击。总体结构如 3.5 所示。

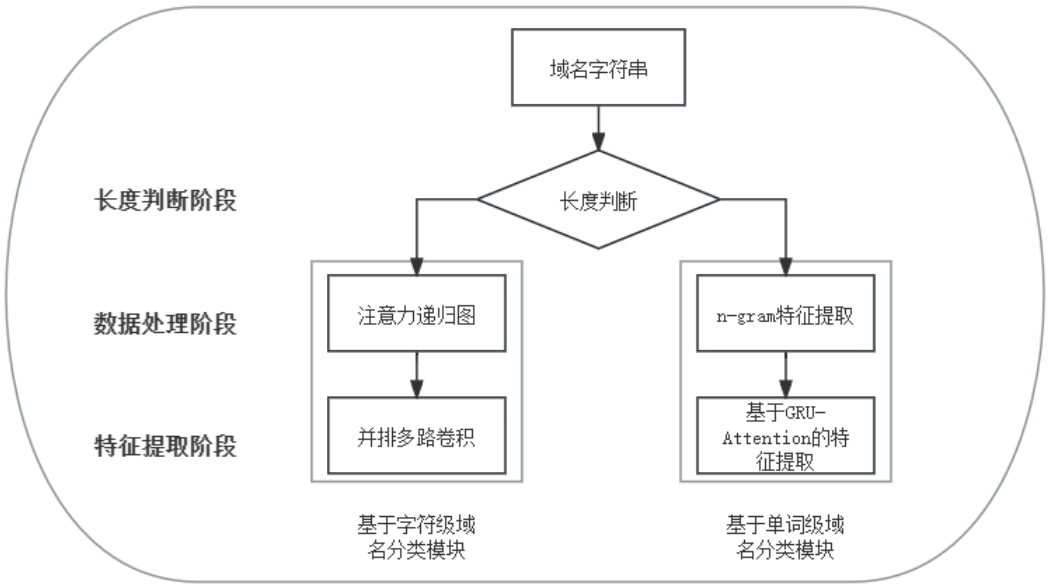


图 3.5 异构 DGA 域名检测模型

3.3.2 域名字符处理规则

一个域名由多个子字符串构成，这些子字符串之间通过点号“.”进行分隔。

每一个这样的子字符串都被称为一个标签。在大多数情况下，传统的二级域名都承载着明确的辨识功能和实用价值。相对而言，由随机数生成算法所产生的 DGA 域名则显得随机性极高，存在时间短暂，通常不含有实质的意义，使其难以与正常域名区分开来。基于上述差异，本节选择以二级域名的长度作为衡量域名长度的标准。

域名是一个字符串，其字符属于可选字符集。由于后续建模要求输入为数组，因此需要对域名进行编码。也就是说，需要一个可选字符和数字之间的映射字典。一般认为，在 Alexa 排名前 100 万域名是良性的。所以本文从 Alexa 下载了所有的顶级域名，提取其 SLD，将所有大写字母转换为小写字母，并将每个 SLD 转换为字符序列。提取其 SLD，删除 SLD 长度为 1 的域名，将所有大写字母转换为小写字母，并将每个 SLD 转换为字符序列[42]。然后计算字符序列中出现的所有字符及其频率。

根据统计，良性域名列表中有 38 个可选字符，包括“0”-“9”、“a”-“z”、“-”和“_”。在这 38 个可选字符中，“_”字符出现频率最低，“e”字符出现频率最高。根据汉字的出现频率，出现频率最低的“_”字对应的数字为 0，出现频率最高的“e”字对应的数字为 37。表 3.1 给出了可选字符与编码数组中数字之间的关联。

表 3.1 可选字符与编码数组中数字的关联

字符	映射值	字符	映射值	字符	映射值	字符	映射值
a	36	b	21	c	28	d	26
e	37	f	18	g	22	h	23
i	34	j	12	k	20	l	29
m	27	n	30	o	35	p	24
q	9	r	32	s	33	t	31
u	25	v	17	w	20	x	13
y	19	z	14	0	8	1	11
2	10	3	6	4	7	5	5
6	3	7	1	8	4	9	2
-	15	_	0				

3.3.3 字符级域名检测模块

字符级域名是指二级域名平均长度在 3~7 个字符之间的域名。这类域名具有字符重叠率高和关键字符比例低的特点，这些特征使得难以从 DGA 域名中手动提取特征。这些特性导致从 DGA 域名中手动提取特征变得更为困难。

因此，本节将探索域名的特征，利用 DGA 域名字符间的相位差，并将这些信息进行图像编码，以表征 DGA 域名。递归图是一种能够反映序列数据内部结构相似性和潜在信息的方法，可在 2D 或 3D 空间中投影和可视化多维相空间数据[43]。即使处理短数据和非平稳数据，也能从中提取有用信息。

为此，本文提出了注意力递归图 ARP，通过增加 DGA 域名中关键字符和增加字符重复性之间的差异，可以提升传统递归图对 DGA 域名的表征能力。在 DGA 域名中，不同类型的字符具有不同的权重，因此 ARP 中的每个像素值也通过加权来计算。通过 ARP 对 DGA 域名关键特征的相空间特征进行差异化表示，突出不同类别 DGA 域名之间的差异，使得分类模型能够快速准确地对不同类别的 DGA 域名进行分类。由于 ARP 可以表示权重计算后各字符之间的相空间距离，因此具有一定的可解释性。

本文以几种常见的 DGA 域名家族为例，探讨了它们的特点。对于这些常见的 DGA 家族，例如 banjori、conficker、nymaim 和 virut，使用日期、社交网络热门话题标签、随机数或字典，甚至每日变化的外汇汇率作为生成域名的种子。即使它们的“算法种子”高度多变，生成的 DGA 域名看似无规则，但其算法逻辑是固定的。例如，对于 antavmu，其顶级域名随机采用“.com”、“.net”和“.org”等常用顶级域名中的一个。其二级域名固定为 7 个字符长度，并以十六进制表示。本文利用递归思想从相空间中挖掘各种差分遗传算法的潜在特征，并将其用 ARP 表示。尽管 DGA 域名各不相同，但本文提出的注意递归图仍然可以将其从一维数据转换为二维图像来表示其相空间特征。

（1）注意力递归图

根据上述描述，可以将 DGA 域名转换为注意力递归图。下面介绍将 DGA 域名转换为 ARP 的原理。

DGA 域名可以被视为一维序列数据，可表示为：

$$DN_x = \{U_0, U_1, \ldots, U_{n-1}\}, x \in \{0,1,2,\ldots,m\} \tag{3-4}$$

其中 $U_i\left(i \in \{0,1,...,n-1\}\right)$ 是域名中第 i 个位置的值。传统的递归图生成时，

需要选择合适的嵌入维数 m、延迟系数 τ、和阈值 ε 来重构一维数据的相空间。虽然常用的嵌入维数选择方法是邻域法，但延迟系数选择方法是平均互信息法。然而，阈值 ε 的选择大多是用 10%，这不可避免地对于递归图上的原始数据的表达不够准确。本文提出了注意递归图，以准确地表示 DGA 域名的相空间特征。首先，ARP 的长度和宽度是 DGA 二级域的长度，为了确保 DGA 域可以被 ARP 完全映射，没有使用阈值 ε 来约束相空间中的距离，以尽可能多地在相空间中呈现 DGA 域。因此，ARP 的长度和宽度为：

$$IS = N - (D-1) \times T \tag{3-5}$$

在公式（3-5）中，IS 表示递归图的长度和宽度，N 表示输入 DGA 域名的二级域名的长度，D 表示输入 DGA 域名的维度，因为域名是顺序数据，并且 D 为 1。T 表示数据输入滑动选择的步骤，本文将其设置为 1，以保证域名表示的准确性。首先，计算输入数据U_i的每个字符数据的关注率：

$$ATT_{U_i} = \frac{Bincound\left(U_i\right)}{N}, i \in (0,1,\ldots,n-1) \tag{3-6}$$

在公式（3-6）中，N 是 DGA 域名的二级域名的长度，*Bincound* 表示计算字符在 DGA 域名中出现的频率。

然后计算它们之间的相空间距离，公式如下：

$$D_{i,j} = \| U_i - U_j \|, i \in \{0,1,\ldots,n-1\}, j \in \{0,1,\ldots,n-1\} \tag{3-7}$$

其中$D_{i,j}$表示 DGA 域名中字符之间的相空间距离，$\|\bullet\|$表示二范数计算，U_i和U_j是每个字符的对应映射值。由于本文没有采用传统的递归图生成算法中的阈值 ε 来权衡相空间距离，而是引入了注意力机制来改进递归图，因此注意力递归图的相空间矩阵为：

$$M_{RP\_x} = \begin{bmatrix} \frac{D_{0,0}}{ATT_{u_0}} & \frac{D_{0,1}}{ATT_{u_0}} & \cdots & \frac{D_{0,n-1}}{ATT_{u_0}} \\ \frac{D_{1,0}}{ATT_{u_1}} & \frac{D_{1,1}}{ATT_{u_1}} & \cdots & \frac{D_{1,n-1}}{ATT_{u_1}} \\ \vdots & \vdots & \frac{D_{i,j}}{ATT_{u_i}} & \vdots \\ \frac{D_{n-1,0}}{ATT_{u_{n-1}}} & \frac{D_{n-1,1}}{ATT_{u_{n-1}}} & \cdots & \frac{D_{n-1,n-1}}{ATT_{u_{n-1}}} \end{bmatrix} \tag{3-8}$$

$$x \in \{0,1,2,\ldots,m\}, i \in \{0,1,\ldots,n-1\}, j \in \{0,1,\ldots,n-1\} \tag{3-9}$$

$D_{i,j}$ 表示输入数据的相空间距离，i 和 j 对应于 DGA 域名中的第 i 个和第 j 个字符。$D_{i,j}$ 的计算如公式（3-7）所示。在获得注意力递归图相空间矩阵之后，对矩阵进行图像编码以显示对应的 ARP。

本文以 DGA 家族中的域名 banjori 为例，说明将域名转换为 ARP 的过程，过程如图 3.6 所示。首先，使用映射字典将域名的每个字符转换为相应的数字（例如，“b”变为 21，“a”变为 36）。此时，banjori 变为对应的纯数字数据，然后根据公式（3-6）和公式（3-7）计算数字数据的关注度比和相空间差。进一步地，根据公式（3-8）得到注意力递归图相空间矩阵，通过对注意力递归图相空间矩阵进行图像编码，可以得到对应的 DGA 域名注意力递归图。

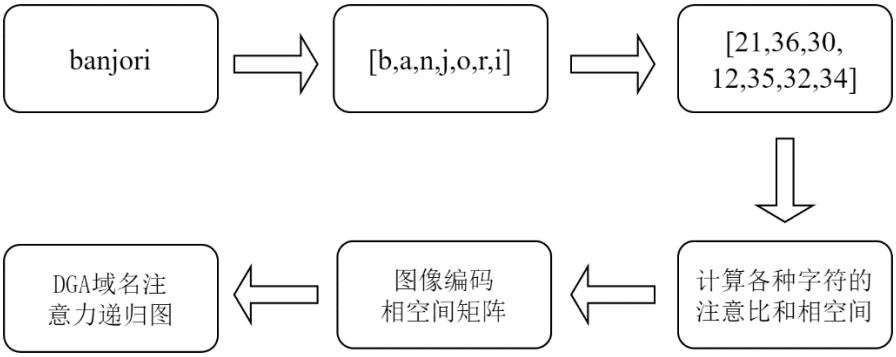


图 3.6 DGA 域名到注意递归图的转换流程图

（2）并行多路卷积

设计了并行多路卷积模块，该模块具有 3 路 CNN 结构来处理注意力递归图，3 向 CNN 结构，从不同的角度提取不同的特征。每种方式的结构都有 3 层，前 2 层用不同大小的卷积层叠加，提取输入词嵌入的局部特征。

第一层是沿着长度方向进行卷积，相当于提取输入序列的 n-gram 特征。

第二层从更大的感受野中提取关键的 n-gram 特征，并降低特征的维数。并行多路卷积不仅能捕获更丰富的特征信息，且提高了模型的表示能力。为了减少信息的损失，在模型中不使用池操作。

第三层是切换归一化（Switch Norm）层[44]，这是一个正则化层。层归一化（Layer Norm）[45]、批归一化（Batch Norm）[46]和实例归一化（Instance Norm）[47]通过切换归一化层进行加权，加权参数在训练过程中学习得到。这种正则化方法可以为神经网络结构的每个归一化层确定合适的正则化操作，从而提高模型的泛化能力。

最后，将并排多路卷积模块的输出叠加在通道维度上。然后全连接分类器判断输入的域名是否为 DGA 域名。全连接分类器由一个全连接层组成 Softmax 层用于对分类结果进行评分。

字符级域名检测模块如图 3.7 所示。

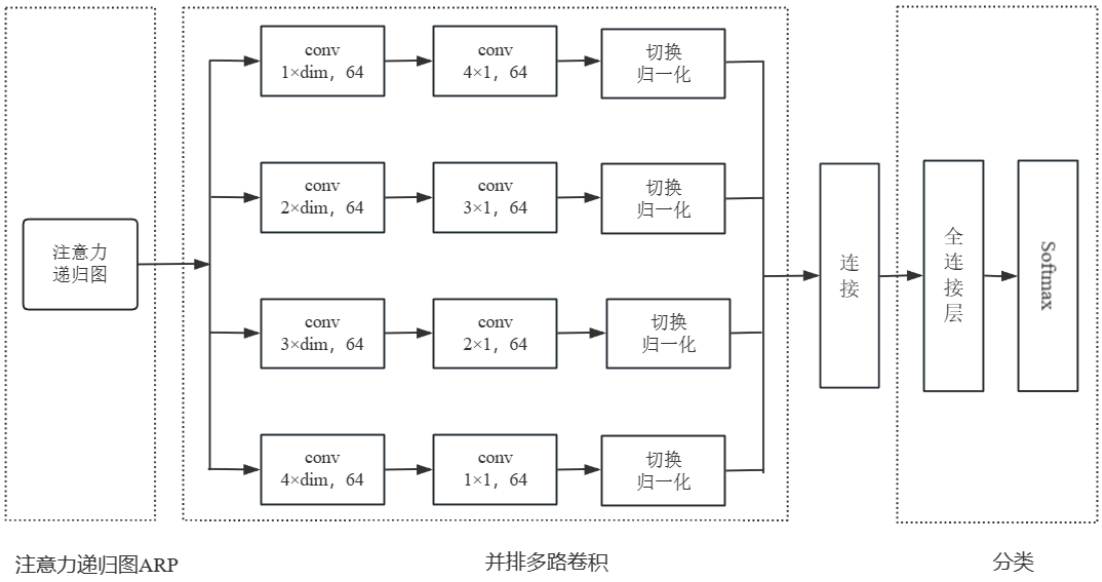


图 3.7 字符级域名检测模块

3.3.4 单词级域名检测模块

为了躲避检测，现在这些恶意域名通常由词典生成，它们选取专有词典中的单词进行组合，以减少字符的随机性并增强迷惑性。为了更有效地对这些恶意域名进行分类，本节对 n-gram 方法进行了改进，以便更准确地表征域名。

（1）右移张量

基于 DGA 的域名和良性域名之间的区别一方面取决于域名组成的字符，另一方面取决于这些字符的排列。右移张量作为域名的表现形式，通过有规律地改变域名中字符的位置，可以形象地描述这种差异。图 3.7 以“taobao”为例，展示了右移张量的过程。图中的尺寸为8×8 。首先，把“taobao”放在矩阵的第一行，然后环右移一个字符，把它放在8×8 矩阵的第二行。以此类推，字符串被环右移 n-1 次（n 是子域的长度），矩阵中没有字符的位置用 0 填充。充分考虑组成域名的字符以及这些字符之间的位置关系。通过逐行和逐列扫描字符串，可以分别获得前向和后向 n-gram 特征。在数据库构建过程中，由于多次复制，不可避免地会引入一些冗余信息。

为了从前向和后向提取输入序列的 n-gram 特征，根据卷积核的结构设计了

不同大小的非对称卷积核。如图 3.8 所示，红色框展示了1×3卷积核操作，而绿色框则展示了3×1卷积核操作。这两种扫描方法共同作用，从而捕捉序列在两个不同方向上的 n-gram 特征。众所周知，n-gram 空间的大小与域名字符空间呈指数增长关系。该结构可以极大地减少特征空间来提取 n-gram 特征。

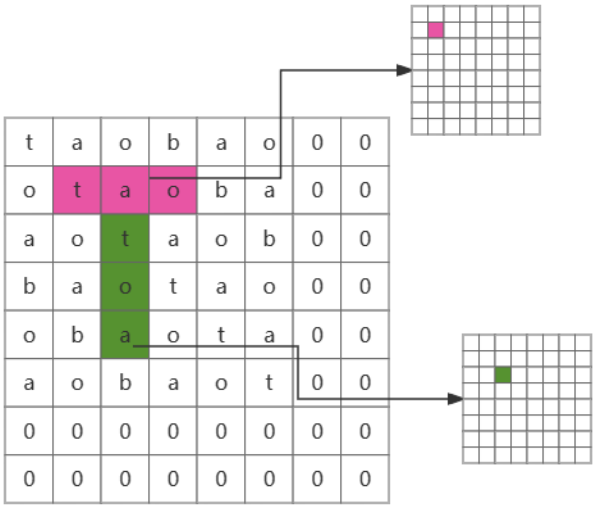


图 3.8 右移张量

(2) 总体架构

单词级域名模块的架构包括三个部分，即 n-gram 特征提取、基于 GRU-Attention 的特征处理、分类。

在进行 n-gram 特征提取的过程中，构建了两种不同的卷积核配置，它们负责从序列的正向与反向分别捕捉 n-gram 的特征信息。在本文中，n 分别取 3、4 和 5。

基于 GRU-Attention 的特征处理。在 n-gram 特征中存在一定量的信息冗余。不同的 n-gram 特征是彼此独立的。为了融合不同的 n-gram 特征，执行1×1卷积。经过压缩编码和 CNN 的维度变换，将数据输入到 GRU 进行特征提取。GRU 的操作流程如下公式所示：

$$
\begin{aligned}
z &= \sigma\left(W^{z}\cdot\left(x^{t}+h^{t-1}\right)\right)\\
\left(h^{t-1}\right)' &= h^{t-1}\odot r\\
h' &= \tanh\left(W\cdot\left(x^{t}+\left(h^{t-1}\right)'\right)\right)\\
h^{t} &= z\odot h^{t-1}+(1-z)\odot h'\\
r &= s\left(W^{r}\times\left(x^{t}+h^{t-1}\right)\right)
\end{aligned}
\tag{3-10}
$$

在公式（3-10）中，z 是更新门，它通过当前输入 x^t 和前一个隐藏状态 h^{t-1}，W^z 是其超参数，并经过一个激活函数 σ 来计算。r 是重置门，用于筛选和重置神经网络上的存储器状态 h^{t-1}，h^{t-1} 是在时间点 $t-1$ 的隐藏状态，W^r 是重置门控超参数。tanh 是双曲正切函数，它可以将输入值压缩到-1 和 1 之间。

注意力机制帮助模型学习域名中的重要关键字，从中获得最重要的特征进行分类。这也减少了模型的计算负荷。在实际应用中，时间的减少也是有意义的。在 GRU 层对数据进行训练后注意力层选择模型中最关键的特征。在 Attention 层之后加入 Dropout 层，用于减少待计算的参数，有助于减少模型的训练时间，但仍能保证基本特征的保留。自注意层是注意的自学习机制，而 Seq 加权注意层是基于权重的注意。当与 GRU 网络结合使用时，这两层的优点和缺点可以相互补充。

经过 GRU 和 Attention 训练过程后，Dense 层拉伸对应于待分类零件类数的计算结果。使用的激活函数为 Softmax，由公式给出，用于预测域名的标签：

$$\mathrm{a}_i = \frac{\exp\left(\mathrm{z}_i\right)}{\sum_{j=1}^{\mathrm{C}} \exp\left(\mathrm{z}_j\right)}, \forall i = 1, 2, \ldots, \mathrm{C} \tag{3-11}$$

在公式（3-11）中，C 是待分类的类数，编号从 1 到 C。在多类分类问题中，C 为待分类的 DGA 僵尸网络族数；集合 z 是 Softmax 函数的输入值，包括 z_i 从 1 到 C，z_i 可以简单地理解为数据属于第 i 类的可能性。因此，z 是 Softmax 激活函数的输入值。

模型各层结构如图 3.9 所示：

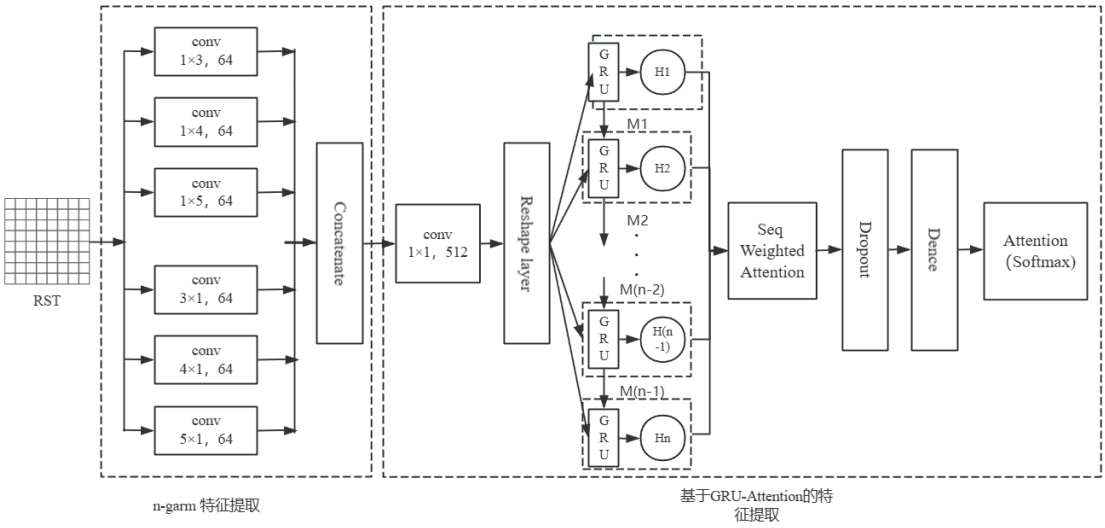


图 3.9 单词级域名检测模块

3.4 本章小结

本章提出了一种创新性的异构 DGA 域名检测模型，该模型利用深度学习与自然语言处理的技术，通过细致的域名长度分析及定制化的特征提取策略，有效地提高了对这两种难以检测域名类型的识别精度。尤其是通过结合 GRU-Attention 机制，模型不仅提升了对数据中关键信息的捕捉能力，还增强了对 DGA 域名长期依赖关系的理解。这种多维度特征分析方法为 DGA 域名检测与分类提供了一种强大的工具，增强了对复杂僵尸网络攻击形态的响应能力。

第 4 章 实验结果与分析

本章对第 2 章中提出的基于生成对抗网络的 DGA 样本生成模型和第 3 章中提出的异构的 DGA 域名检测模型进行实验验证，通过实验结果对比分析，验证本文提出的方法在进行 DGA 域名检测和分类的准确性与可靠性。

4.1 基于 GAN 的 DGA 域名生成方法实验

该节的主要目的是开发出能够贴合目标域名的样本。首先，本节介绍了实验环境和生成域名的一些实验细节，并评估了生成域名和真实域名的相似性。然后，利用公开数据集进行分类效果评估，并通过多种机器学习算法验证生成域名的有效性。最后通过和其他生成模型做对比实验，验证本文模型的优良性。

4.1.1 实验环境和实验设计

本文使用的实验平台及配置信息如表 4.1 所示。

表 4.1 实验平台及配置信息

参数名称	配置
操作系统	Window11
内存	8GB
GPU	GeForce RTX 3060
CPU	AMD Ryzen 75800
显存	24GB
编程语言	Python 3.9.13
深度学习框架	TensorFlow2.3
机器学习平台	Weka3.8

本节实验主要分为三部分，一是 DGA 域名样本的生成与验证实验，二是评估生成的 DGA 域名的有效性，通过对比实验，判断是否可用于小样本 DGA 域名数据增强。三是与其他域名生成模型作对比，验证本文模型的优良性能，实验设计如下：

（1）生成域名分析：使用 NWGAN 模型进行对抗样本生成。验证为检测模型提供高质量的恶意域名样本的可能性。通过比较真实样本和生成样本的域名长度和字符间的转换等特征，验证生成样本的有效性。

（2）生成域名评估：使用朴素贝叶斯、随机树、随机森林和 J48 决策树分类器进行分类。将分类器在真实样本上的检测结果作为基准，与分类器在生成样本上的检测结果进行比较。通过计算分类器在生成样本上的正确率、错误率、精确率和 F1 指标，评估生成样本的质量和对分类器性能的提升效果。

（3）与其他域名生成模型作对比：使用域名检测模型检测生成的域名样本，通过观察这些检测模型对于生成域名的识别性能，来评估生成器的能力。

4.1.2 数据集

Conficker.C 恶意域名数据集和 Alexa 良性域名数据集是全球公认并广泛使用的数据集。本节实验数据集选取了近 50 万个 Conficker.C 恶意域名，这些域名均来源于近期的网络安全事件和报告，具有较高的时效性和代表性。同时，从 Alexa 最新排名中选取了排名前 100 万的良性域名。这些良性域名不仅知名度高，而且用户访问量大，因此作为对照数据集具有很强的说服力。

4.1.3 模型评价指标

选取准确率（Accuracy）、错误率(Error Rate)、精确率（Precision）、F1 值作为生成域名模型的评价指标。四者公式如下：

$$\text{Accuracy} = \frac{TP + FP}{TP + TN + FP + FN} \tag{4-1}$$

$$\text{Error Rate} = \frac{FP + FN}{TP + TN + FP + FN} \tag{4-2}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4-3}$$

$$F1 = \frac{\text{Precision} * \text{Recall} * 2}{\text{Precision} + \text{Recall}} \tag{4-4}$$

其中：

TP（真正例）：表示模型预测为恶意域名且实际也是恶意域名的数量，这个指标反映了模型正确识别恶意域名的能力。

TN（真负例）：表示模型预测为正常域名且实际也是正常域名的数量，这个指标反映了模型正确识别正常域名的能力。

FP（假正例）：表示模型预测为恶意域名但实际是正常域名的数量。这个指标反映了模型错误地将正常域名识别为恶意域名的情况，产生了误报。

FN（假负例）：表示模型预测为正常域名但实际是恶意域名的数量。这个指标反映了模型未能正确识别恶意域名的情况，产生了漏报。

TP 和 TN 两个指标越高，说明模型的分类效果越好。FP 和 FN 两个指标越低，说明模型的分类效果越好。通过综合这些指标，可以更全面地评估模型在识别恶意域名和正常域名方面的性能，从而优化模型，提高分类效果。

4.1.4 生成域名分析

在生成对抗网络的训练流程中，必须谨慎处理鉴别器的训练进度。如果鉴别器过早地达到最优状态，将会导致梯度太小，进而使得生成器的参数更新和优化变得困难。同样，如果鉴别器的性能过于低下，它将向生成器传递错误的梯度信息。在这项实验中采用了 RMSprop 优化算法，学习率设置为 0.00005，并批处理大小为 64。每个训练周期开始时，对鉴别器网络进行两次训练，然后冻结其参数，训练生成器一次。每经过 500 次迭代，将输出数据进行一次解码，并将解码后的域名保存在文本文件中。实验中将目标域名设定为合法域名。在整个实验过程中，每 500 次迭代后，利用生成器生成了一些样本，用于性能评估。表 4.2 展示了部分生成的样本：

表 4.2 生成域名不同迭代阶段样本

迭代次数	样本
500	ccccc
1000	gologoo
1500	Tooootlottttotltt
2000	Caaatattsrrrocats
2500	voccttmroctts
3000	mottigmgsatidm
3500	potrogmataihpskbtrotk
4000	carudesftarolus

从上述表格中可以看到，在生成对抗网络训练的初期，生成的样本中还存在一些重复的字符，但已经能够观察到一些类似自然语言的特点。随着迭代的深入，当达到 3000 次迭代后，生成的数据逐渐转变为可拼读的字符串，显示出模型在生成具有语言特性的文本方面的能力有了显著提升。最终生成了一万个样本作为测试集，并将其标记为 GANDomain。接下来，将继续对 GANDomain 测试集中的样本进行一系列的分析和测试，以探究模型在生成恶意域名方面的实际效果和潜在应用。

如图 4.1 所示，因为采用了 n-gram 字典和 WGAN 相结合的方法，生成域名避免了大量字符重复的情况。从域名长度和字符间的转换等外观特征来看，成功地生成了与目标域名高度相似的生成域名。

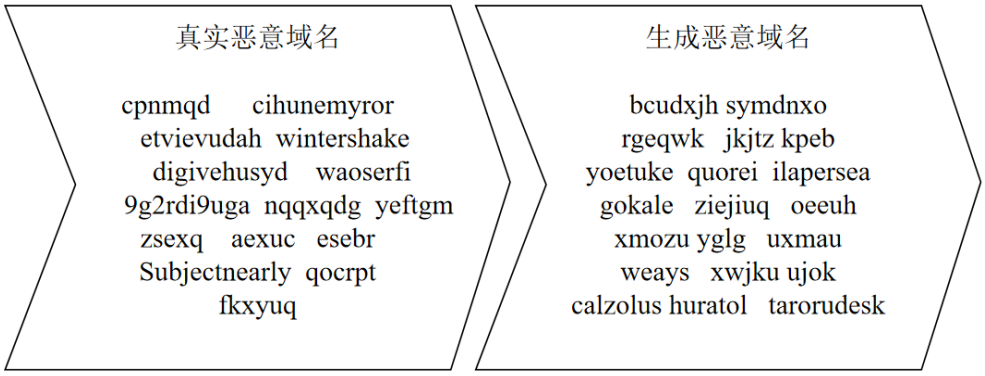


图 4.1 真实恶意域名与生成恶意域名示例

图 4.2 中，展示了通过分析单个字符出现的频率，来比较和区分真实样本与生成样本的不同。图中的白色圆圈代表的是真实样本中各个字符的平均频率，而黑色的菱形代表的则是生成样本中字符的平均频率。可以看出，两种样本的字符频率在不同字符间波动，但总体趋势相对一致。因此，表明生成样本与真实样本具有相似性。

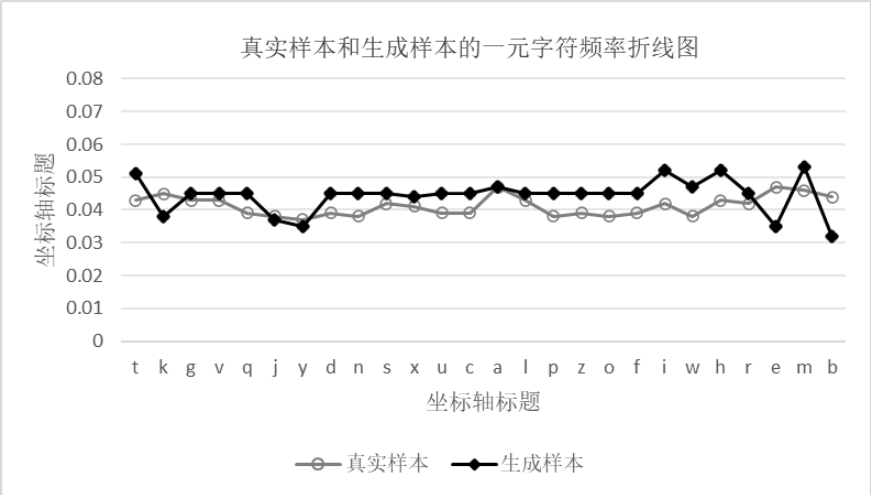


图 4.2 真实样本和生成样本的一元字符频率折线图

4.1.5 生成域名评估

生成域名的一个主要目的是对目标域名家族进行数据增强，特别是对于样本量有限的家族。为验证生成域名的有效性，本节采用了包括朴素贝叶斯、J48 决策树、随机森林和随机树在内的多种机器学习分类器进行比较分析。

实验中使用了三个数据集：

① 放有 10000 个 Conficker.C 中的恶意域名和 10000 个 Alexa 的良性域名。

② 用本文模型生成的恶意域名和 Alexa 的良性域名，选择相同的数据特征进行分类，其结果和①进行比较。

③ Conficker.C 恶意域名、生成恶意域名和 Alexa 的良性域名进行混淆分类，分类结果与①和②比较。

表 4.3 Conficker.C 恶意域名与 Alexa 良性域名分类结果

分类器	准确率	错误率	精确率	F1
朴素贝叶斯	0.985	0.015	0.985	0.985
J48	0.984	0.016	0.986	0.984
随机树	0.986	0.014	0.986	0.986
随机森林	0.988	0.012	0.988	0.988

表 4.3 展示四种分类器对恶意域名与良性域名分类的结果。可以看出，所有分类器的准确率都很高，接近或超过 98.5%，这表示这些分类器对于数据集的预测非常准确。误报率非常低，这意味着将良性域名错误地分类为恶意域名的情况非常少。F1 分数也很高，接近 98.5%以上，这表明分类器在保持高召回率的同时也保持了较高的准确率。分类结果表明，所使用的特征提取和分类算法非常适合于恶意域名检测任务。此表结果为下面各数据集的分类提供了基准。

表 4.4 生成的恶意域名与 Alexa 良性域名（1:1）

分类器	准确率	错误率	精确率	F1
朴素贝叶斯	0.963	0.037	0.965	0.963
J48	0.952	0.048	0.954	0.952
随机树	0.983	0.017	0.983	0.983
随机森林	0.981	0.019	0.981	0.981

表 4.5 生成的恶意域名与 Alexa 良性域名（1:2）

分类器	准确率	错误率	精确率	F1
朴素贝叶斯	0.958	0.042	0.959	0.959
J48	0.945	0.055	0.946	0.946
随机树	0.981	0.019	0.981	0.981
随机森林	0.988	0.012	0.988	0.988

以表 4.3 提供的分类器对真实恶意域名与良性域名的分类结果作为基准，在

表 4.4 和表 4.5 中，使用生成的恶意域名样本，分类器依然保持了高准确率，表明生成的样本质量与真实样本足够相似，可以有效用于训练分类模型。表 4.4 中保持了与表 4.3 相同的比例，而在表 4.5 中，恶意域名与良性域名的比例改变为 1:2，模型的准确率和 F1 分数略有降低，但仍然保持在 94%以上，表明模型对于恶意域名的检测依旧有效，即便在负样本较少的情况下。证明了生成的域名样本对于训练高性能的恶意域名检测模型是有效的

表 4.6 混淆样本（1:1）与 Alexa 良性域名分类结果

分类器	准确率	错误率	精确率	F1
朴素贝叶斯	0.972	0.028	0.972	0.972
J48	0.963	0.037	0.965	0.963
随机树	0.986	0.014	0.986	0.986
随机森林	0.991	0.009	0.9991	0.9991

表 4.7 混淆样本（2:1）与 Alexa 良性域名分类结果

分类器	准确率	错误率	精确率	F1
朴素贝叶斯	0.971	0.029	0.971	0.971
J48	0.963	0.037	0.964	0.963
随机树	0.977	0.023	0.977	0.977
随机森林	0.989	0.011	0.989	0.989

表 4.6 展示了生成恶意域名、真实恶意域名的混合样本和 Alexa 良性域名进行分类，表 4.7 中，混合比例从 1:1 调整为 2:1。从这两个表中可以看出，分类准确率仍然很高，即使在生成样本更多的情况下，分类器的表现也没有显著下降。这表明生成的域名在特征上足够接近真实的恶意域名，避免被分类器区分。因此，这些生成域名可以用于增强机器学习模型的恶意域名检测和训练能力。

从上面三组对比实验综合分析，以第一组实验为基准，后续实验中增加生成恶意域名并分别与良性域名和混淆域名进行分类比较。同时，对每组实验中不同数量和比例的样本集进行了深入分析。所有结果均显示出高分类准确率，证实了生成恶意域名的有效性。

为了更加清晰的观察数据增强效果，选择两个样本量较少的 DGA 家族进行增强实验。对于样本不足的可拼读 DGA 家族如 symmi 和 suppobox，由于训练数据不足，传统的检测模型往往难以有效地对这些 DGA 域名进行分类，从而导致较低的检出率。为了解决这个问题，采用本文提出的基于 n-gram 字典和

WGAN 的 DGA 样本生成模型来生成额外的样本。具体来说，将 symmi 和 suppobox 家族的域名作为目标域名，分别生成了 30000 个样本。这些生成的样本随后与真实样本混合，使得新的训练数据样本数分别是原始样本数的 8.25 倍和 10.85 倍。这样的数据增强策略有效地扩充了训练集，使得模型能够学习到更多关于这两个 DGA 家族的特征。为了评估经数据增强后小样本 DGA 家族检测效果的提升，将训练集更新为新合成的数据集。本研究复用了上面的实验方法对检测模型进行再次训练，以此来验证数据增强对提高检测模型对小样本 DGA 家族成功识别率的影响。实验结果如图 4.3 所示。其中纯色是原数据集分类结果，斜线是对应的增强数据集分类结果。

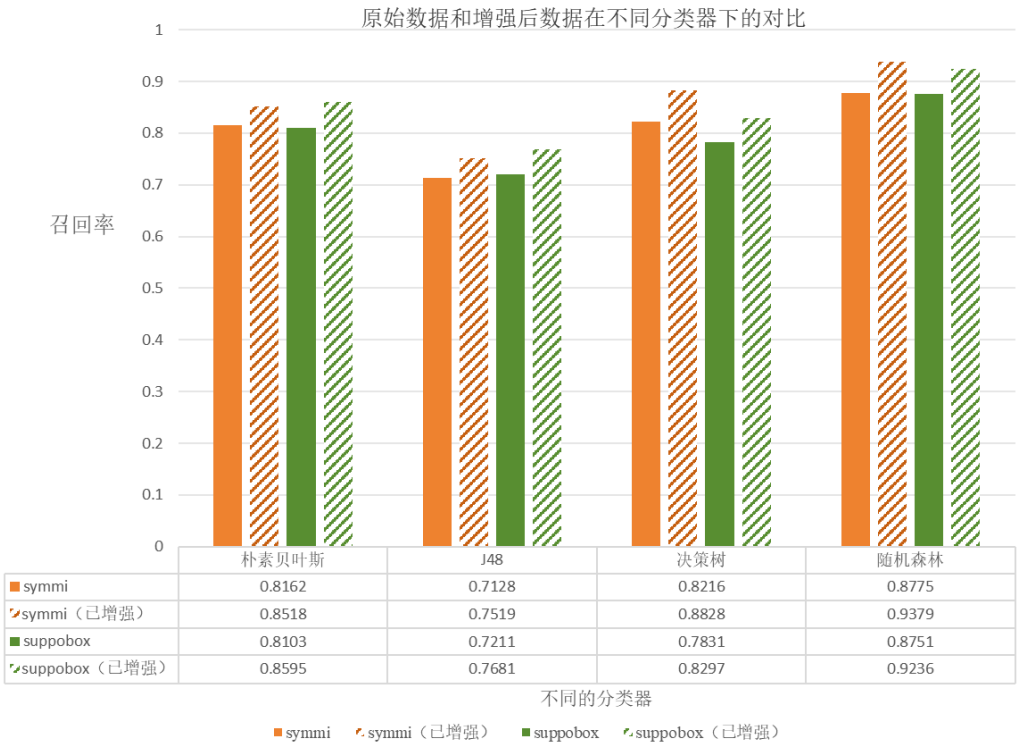


图 4.3 数据增强后 DGA 域名家族二分类召回率

数据增强后，各个分类模型的性能普遍得到增强，尤其在 Symmi 家族的识别上，由于拥有更多初始样本量，模型召回率有显著提升，例如决策树和随机森林的召回率平均上升了 6 个百分点。这种提升得益于生成模型在更充足的原始数据训练下，能更准确地模拟目标数据特性，相应地提升了训练数据集的代表性和检测模型的分类性能，证明了生成域名的有效性。

综上所述，数据增强技术通过扩充训练集、提高数据质量，有效地提升了 DGA 域名检测分类模型的性能。这一结果为 DGA 域名的检测提供了有力的支持，证明了数据增强技术在处理小样本问题时的有效性。

4.1.6 对比实验

本节中，选择了 DomainGAN[48]、DeepDGA[49]、和 CL-GAN 三种不同的域名生成器，以及本文提出的生成器 NWGAN，进行对比试验，旨在评估各生成器的性能。其中，DomainGAN 使用生成对抗网络(GANs)的三种不同变体，通过使机器学习算法更难检测域来改进域生成型。详细介绍了训练这种基于字符的生成对抗网络的几个挑战的解决方案。DeepDGA 的深度学习架构始于一个域名自动编码器，在竞争性地重新组装，具有新颖的神经架构和训练策略以提高收敛性。如果检测模型对于某个生成器产生的域名样本的检测性能较差，说明该生成器在构造难以识别的恶意域名方面表现更为出色，从而证明其生成能力更强。从图 4.4 可以看出，良性域名的长度集中在 11-20，而 DGA 域名的长度集中在 11-30。

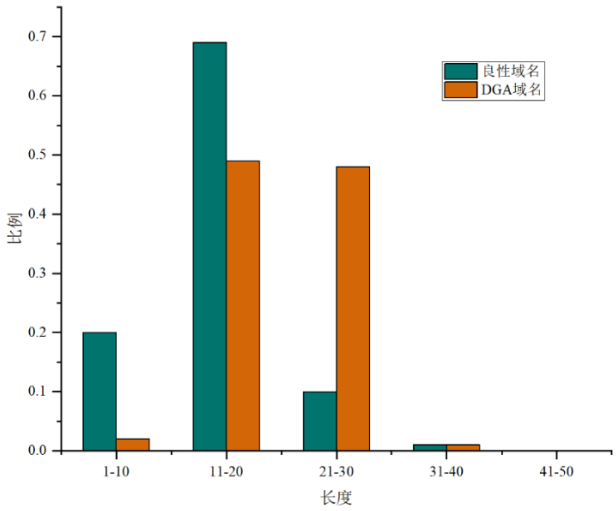


图 4.4 域名长度的分布

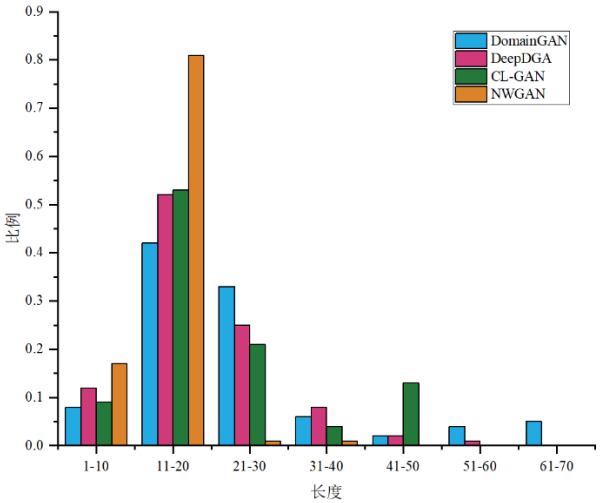


图 4.5 由四种生成器生成的域名的长度

图 4.5 显示了由不同方法生成域名的长度。从图 4.5 可以看出，生成域名的长度大部分集中在 11-20，并且 NWGAN 的比例最高，因此 NWGAN 的生成域名比 DomainGAN、DeepDGA、和 CL-GAN 更接近良性域名。

全部使用基于 GAN 的模型的检测器评估生成器的能力没有意义，因为相同的基础架构可能导致评估结果有偏差。所以，选择两个非基于 GAN 的模型 LSTM.MI 和 HAGDetector，同时选择基于 GAN 的模型 DomainGAN、DeepDGA、CL-GAN 和本文模型 NWGAN 作为检测器，来测试不同生成器生成的样本。因为所有检测器都是根据生成的样本进行评估的，所以检测结果的精确率都为 1，召回率越低，生成器的强度越大。实验结果如表 4.8 所示。

表 4.8 四种生成器召回率对比

域名生成器	Recall					
	LSTM.MI	HAGDetector	DomainGAN	DeepDGA	Khaos	NWGAN
DomainGAN	0.42	0.53	-	0.46	0.41	0.62
DeepDGA	0.39	0.52	0.42	-	0.44	0.61
CL-GAN	0.41	0.45	0.38	0.4	-	0.53
NWGAN	**0.31**	**0.42**	**0.35**	**0.35**	**0.38**	-

从表 4.8 中得出，NWGAN 生成器生成的样本导致检测器的表现最差，比如 NWGAN 生成的样本使得基于 LSTM.MI 的检测模型的召回率只有 0.31，而 DomainGAN、DeepDGA 和 CL-GAN 生成的样本使 LSTM.MI 的召回率分别为 0.42、0.39 和 0.41。这表明 NWGAN 在生成难以识别的恶意域名方面比其他 GAN 模型表现更好。证明了本文模型优良性能。

4.2 异构 DGA 域名检测模型方法实验

4.2.1 实验设计

本节实验主要分为三部分，一是验证使用本文提出的注意力递归图对短域名表征能力，二是 DGA 域名二分类实验，三是 DGA 域名多分类实验，通过对比实验，判断是否对于短域名和基于单词表的域名检测与分类达到良好的效果。实验设计如下：

（1）将多个 DGA 域名转换为注意力递归图，能够提取更多具有代表性的特征，提高了对短域名表征能力。

（2）使用异构 DGA 域名检测模型进行二元分类，并与其他知名的方法在

短 DGA 域名检测、基于单词的 DGA 域名检测和完全未知的 DGA 域名检测三个方向进行比较，验证本文模型的优良性。

（3）使用异构 DGA 域名检测模型进行多分类实验，验证在字符级域名分类检测和单词级域名分类检测优良性，最后与其他知名的模型进行混合域名实验比较，验证本文模型的优良性。

4.2.2 数据集

数据集分为两部分：正常域名 B-DN 和 DGA 域名 DGA-DN。正常域名主要来自 Alexa。而 DGA 域名则来自多个来源，包括 360 网络实验室、DGArchive 和第二章介绍的 NWGAN 网络生成样本。

为了使模型对不同长度的样本具有相同的检测能力，构建了基于长度平衡的数据集，这意味着对于每一个长度的域名，正常样本和恶意样本的数量是大致相同的，确保模型的检测能力在不同长度的域名上是一致的。根据域名的长度，数据集被分为 2 个组：短域名数据集（CDN，3-7 个字符），长域名数据集（LDN，7 个字符以上），如表 4.9、4.10 所示。

表 4.9 短域名数据集 CDN

类型	家族名称	最小长度	最大长度	样本数	样本示例
B-DN	-	3	7	200000	-
DGA-DN	qsnatch	3	7	122,918	qsn1tx
	conficker	3	7	22,916	cfnkt5
	nymaim	3	7	22,916	nym8im
	virut	3	7	6,250	vi5rtu
	proslikefan	3	7	6,250	prlf3n
	pykspa	3	7	6,250	pks2p
	GANDomain	3	7	12500	yglq

表 4.10 长域名数据集 LDN

类型	家族名称	最小长度	最大长度	样本数	样本示例
B-DN	-	8	35	400000	-
DGA-DN	pykspa_v1	8	15	40000	aaamiaiq
	gameover	16	31	40000	1d7ywjlejzvmktu
	ranbyus	14	17	30000	tfholjtspefhoxope

表 4.10（续表）

类型	家族名称	最小长度	最大长度	样本数	样本示例
	murofet	11	34	10000	yhsqkploqshhjwbl
	locky	8	17	10000	urmrhedeslqtsq
	ramnit	8	19	30000	imboajaksftmyk
	symmi	8	15	10000	anboinkut
	tinba	9	12	40000	nvfowikhevmy
	simda	8	11	30000	dikoniwudim
	banjori	8	26	30000	earnestnessbiophy
	rovnix	18	18	40000	h53bxo2qz45n6io7um
	emotet	16	16	40000	affqvugewqpbcbic
	necurs	8	21	10000	kqeykmhnmgfmsix
	shiotob	8	15	10000	shiotob
	GANDomain	8	30	30000	potrogmatai

4.2.3 模型评价指标

使用五个标准指标对模型进行评估，即精确率（Precision），准确率（Accuracy），召回率（Recall），F1-Measure（F1），和假阳性率(FPR)。

五个指标计算公式分别如下所示：

$$\text{Accuracy} = \frac{TP + FP}{TP + TN + FP + FN} \tag{4-5}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4-6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4-7}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4-8}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{4-9}$$

在本实验中，TP、TN、FN 和 FP 的含义与上一个实验相同。同时，高精确率意味着分类器产生较少的假阳性结果，而高召回率指示假阴性的数量较少。因此，在现实场景中被接受和部署的分类器应该具有高准确率、高精确率和高召回率。此外，为了减少误报，FPR 应当保持在低水平。这些指标共同决定了

分类器在实际应用中的效果。

4.2.4 ARP 消融实验

本实验将多个 DGA 域名转换为相应的注意递归图。图 4.6 为 banjori、nymaim、pykspa、virut 等不同类型的短 DGA 域名转换后的注意递归图示例，并进行热图处理，以便于观察各种注意递归图的间隔。

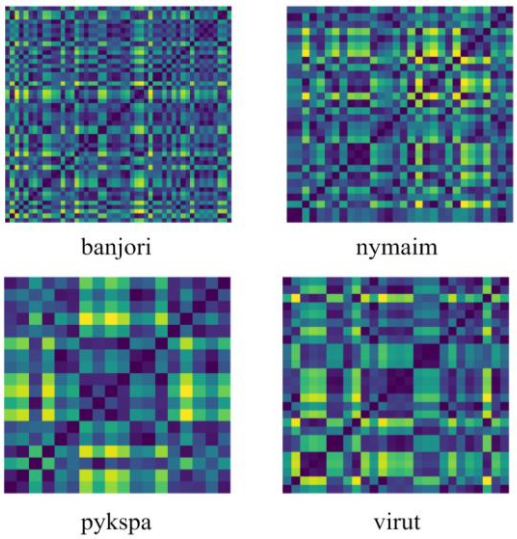


图 4.6 不同类别样本的注意力递归图

从这 4 个 ARP 来看，不同类别的 DGA 域名生成的递归图纹理特征差异较大。注意力机制使域名中频率较低字符的相空间距离像素值被突出显示，以提高不同类别的 ARP 之间的差异。域名中出现次数较少的字符所得到的相空间距离，与四张热图处理的 ARP 的浅色部分相对应。相比之下，注意递归图中较暗的部分是出现频率较高字符的相空间距离。

表 4.11 ARP 消融实验结果

检测模型	Accuracy			
	banjori	nymaim	pykspa	virut
含 ARP	0.971	0.963	0.977	0.989
不含 ARP	0.933	0.924	0.938	0.942

表 4.11 的 ARP 消融实验结果表明，将 DGA 域名转换为递归图实现不同 DGA 的多类分类是可行的。明显提高了分类准确率，ARP 为提取和编码 DGA 域名的相空间特征以输出高精度的分类结果提供了前提条件。

4.2.5 二分类实验结果

本节评估了本文提出的方法在多个数据集上区分恶意域和良性域（即二元分类）的能力。使用 Precision、Recall、Accuracy、FPR 和 F1 的标准分类指标来评估训练过的分类器的性能。

一共进行了三个实验，具体如下：

（1）与其他知名方法比较。复制了 4 种最知名的 DGA 检测方法，分别使用第 4.2.2 节的数据集进行基于短域名的、基于长域名的和混合域名的二分类实验，验证域名长度对模型的检测性能的影响。

（2）基于单词表的 DGA 域名检测。收集了多个基于单词的 DGAs，并形成了一个新数据集进行二分类实验，以验证模型检测基于字词的样本的能力。

（3）完全未知的短 DGA 域名检测。从训练数据集中去掉目标 DGA 族。因此，该模型不具有关于遗漏 DGA 域名的先验知识。这样验证了检测未知样本的能力。

（1）与知名的方法比较

在实验中，与其他文献中的知名方法进行了比较，具体如下：

FANCI：使用随机森林将不存在的域名分为良性和恶意域名。这个分类器使用一些与语言无关的特性。

n-CBDC[50]：结合了 n-gram 和深度卷积神经网络，以端到端的方式运行，不需要手工提取的特征或域名上下文信息；它只需要输入域名本身，就可以自动估计域名被 DGA 生成的概率。在下面的实验中选择了 n=2。

TF-IDF[51]：基于 TF-IDF 的 LR 模型使用 n-grams 的 TF-IDF 值作为 Logistic Regression(LR)算法中域名分类的特征。

ResNet[52]：此模型是神经网络领域中具有较大影响力的经典深度神经网络。在实验中用 PyTorch 实现了 ResNet。

将上述相关方法与本文的方法进行了比较。

在对比实验中，从列出的 5 个指标进行对比分析，结果如表 4.12 所示。从对比结果可以看出，其他方法对超短域名的处理效果较差。本文的方法受样本长度的影响较小，检测准确率保持在 90%以上。此外，从表 4.13 中可以得出，随着域名长度的增加，各方法的检测性能都有所提高。因此，对比实验的结果证明，域名长度对模型的检测性能影响较大。

表 4.12 短域名数据集 CDN 上与知名方法的比较结果

方法	Precision	Accuracy	Recall	F1	FPR
FANCI	0.7305	0.7443	0.7443	0.7441	0.2557
n-CBDC	0.0000	0.5000	0.4999	0.3333	0.4999
TF-IDF LR	0.7521	0.7557	0.7557	0.7557	0.2443
ResNet	0.8412	0.8605	0.8605	0.8604	0.1395
本文模型	**0.9436**	**0.9444**	**0.9444**	**0.9444**	**0.0556**

表 4.13 长域名数据集 LDN 与知名方法的比较结果

方法	Precision	Accuracy	Recall	F1	FPR
FANCI	0.8992	0.8921	0.8921	0.8921	0.1079
n-CBDC	0.9509	0.9669	0.9669	0.9669	0.0331
TF-IDF LR	0.9973	0.9971	0.9971	0.9971	0.0029
ResNet	0.9676	0.9794	0.9794	0.9794	0.0206
本文模型	**0.9927**	**0.9901**	**0.9901**	**0.9901**	**0.0100**

表 4.14 合并数据集与知名方法的比较结果

方法	Precision	Accuracy	Recall	F1	FPR
FANCI	0.8992	0.8921	0.8921	0.8921	0.1079
n-CBDC	0.9440	0.9218	0.9218	0.9218	0.0782
TF-IDF LR	0.9509	0.9669	0.9669	0.9669	0.0331
ResNet	0.9486	0.9450	0.9450	0.9450	0.0549
本文模型	**0.9858**	**0.9875**	**0.9876**	**0.9876**	**0.0124**

根据上面三张表的结果来看，本文模型在所有的数据集中都表现最好，特别是在短域名的检测和分类上实现了较大的提升。同时本文模型在所有数据集上的准确率和 F1 分数都很高，这表明模型在恶意流量识别方面既准确又可靠。此外，本文模型的误报率也是最低的，这意味着将正常域名误判为恶意的情况很少。因此，本文提出的异构 DGA 域名检测模型在检测短域名方面达到了很好的效果。

（2）基于单词表的 DGA 域名检测

一些经过特殊设计的基于单词表的 DGA 域名与良性域名具有类似特征，使得检测模型难以区分。例如，matsnu 有两个词典，一个用于动词，另一个用于

名词，域名的生成取决于种子，始于动词或名词，并交替从两个词典中选择单词，直到域名长度超过 24 个字符。nymaim2 通过连接两个词典中的词构成域名，有时不使用分隔符，有时使用“-”。rovnix 使用《美国独立宣言》作为词典。gozi 是 rovnix 的变种，其词典来自多种公共领域材料，随机从各种文本中选词并连接成域名，长度在 12 至 23 个字符之间。suppobox 包含一个有 384 个词的词典，随机选取并连接两个词。

为了验证模型检测基于单词的 DGA 域名样本的能力，将基于单词的 DGA 域名与正常域名相结合，构建了一个用于二元分类实验的新数据集。表 4.15 提供了数据集的概览。

表 4.15 二元分类实验的新数据集样本分布

域名类型	域名家族	样本数
B-DN	-	100000
DGA-DN	ranbyus	10000
	murofet	10000
	ramnit	10000
	symmi	10000
	tinba	10000
	simda	10000
	banjori	10000
	rovnix	10000
	emotet	10000
	necurs	10000

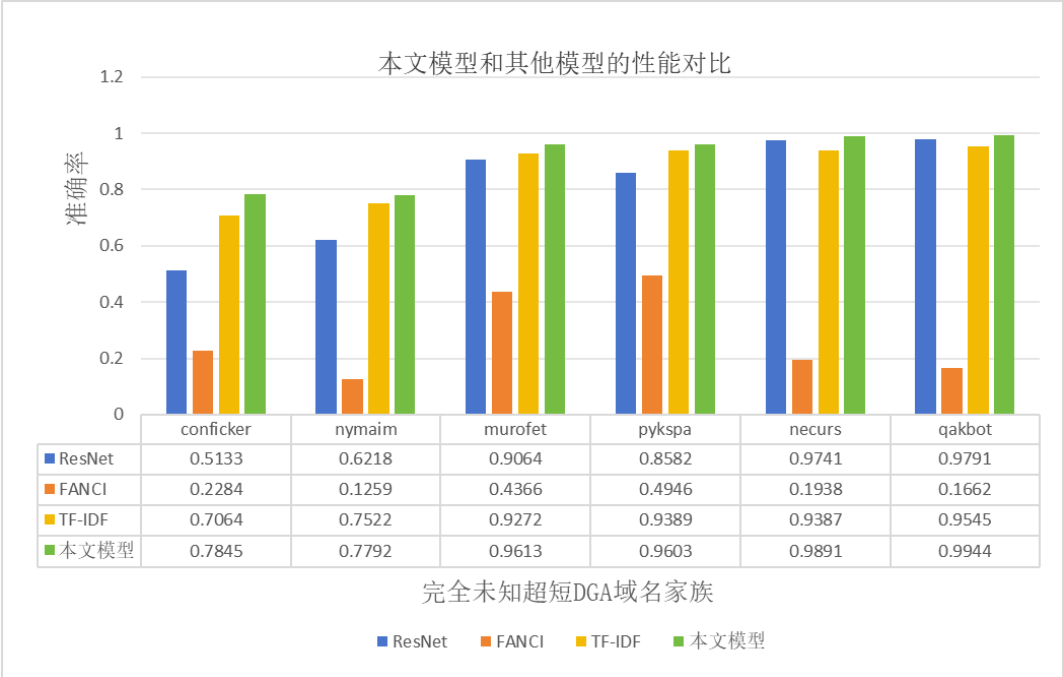
表 4.16 二元分类实验结果

方法	Precision	Accuracy	Recall	F1	FPR
FANCI	0.6920	0.7300	0.7300	0.7105	0.3080
n-CBDC	0.0000	0.5000	0.5000	0.3333	0.5000
TF-IDF LR	0.8119	0.8178	0.8178	0.8178	0.1822
ResNet	0.8487	0.8494	0.8494	0.8494	0.1506
本文模型	**0.8685**	**0.8685**	**0.8784**	**0.8784**	**0.1216**

检测结果显示在表 4.16 中。可以看出，本文的方法可以区分测试家族的基于单词表的域名和正常域名，检测率超过 87.84%。这些结果展示了本文方法在区分基于单词表的域名和正常域名方面的有效性。

（3）完全未知的 DGA 域名检测

为了验证本文方法检测未知的 DGA 域名的能力，用于验证的目标 DGA 域名家族在训练期间被忽略。因此，目标 DGA 域名对模型来说是完全未知的。然后，尝试在没有任何先验知识的情况下，是否可以正确预测目标 DGA 域名。



	conficker	nymaim	murofet	pykspa	necurs	qakbot
ResNet	0.5133	0.6218	0.9064	0.8582	0.9741	0.9791
FANCI	0.2284	0.1259	0.4366	0.4946	0.1938	0.1662
TF-IDF	0.7064	0.7522	0.9272	0.9389	0.9387	0.9545
本文模型	0.7845	0.7792	0.9613	0.9603	0.9891	0.9944

图 4.7 本文模型和其他模型的性能对比

首先，按照指定的长度区间取 50000 个样本进行模型训练，然后使用遗漏 DGA 域名来验证模型的性能，在这里选择了 conficker、nymaim、murofet、pykspa、necurs 和 qakbot 六个未知的 DGA 域名家族进行检测。实验结果如图 4.7 所示，其中纵坐标表示检测的准确率，横坐标是六个未知的 DGA 域名家族。从图 4.7 可以看出，可以看出，本文模型在多个未知 DGA 域名家族的检测中，普遍展现出高准确率。特别是在 necurs 和 qakbot 家族中，本文模型的性能超过了 0.98 的准确率，显示出了优于其他模型的识别能力。这些结果表明，本文提出的模型对于 DGA 域名检测具有强大的实用性和有效性。

4.2.6 多分类实验结果

本节评估了本文提出的方法进行多分类任务的能力，即将 DGA 域名分为多

个类别，以便进一步分析和处理。通常情况下，DGA 域名的类别与不同的恶意软件家族或恶意活动相关联，因此将其进行分类可以帮助安全研究人员更好地理解和应对恶意网络活动。

一共进行了两个实验，具体如下：

（1）分别对字符级和单词级域名进行多分类检测。根据字符级和单词级域名的特征选取了多个 DGA 域名家族对模型执行了多分类任务，以验证字符级域名模块和单词级域名模块的检测能力。

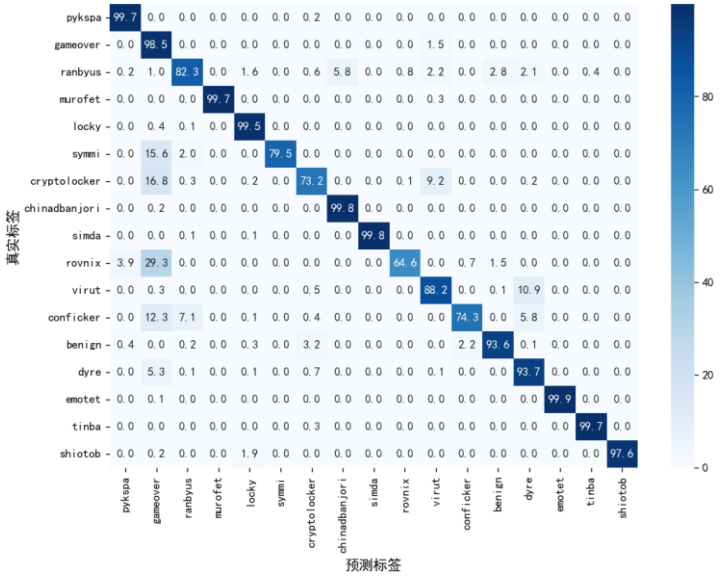
（2）混合分类并与其他优秀模型比较。选取多个具有代表性的短域名家族和基于单词表的域名家族混合，构成了一个用于检测的数据集，并与其他优秀模型进行比较，以验证本文整体模型检测能力及其卓越性。

为了对模型分类性能进行精准评估，本节采用了归一化混淆矩阵的方式。混淆矩阵作为一种表格形式，主要用于衡量分类模型的效能，其中，行对应预测标签，列对应真实标签，对角线上的数值代表正确预测的实例数量，而对角线以外的数值表示预测错误的情况。深色的格子通常表示数量较多，而浅色则表示数量较少。在混淆矩阵的可视化过程中，有时会对每行数据进行归一化处理，这样就能更为清晰地洞察模型在不同类别上的表现，而不必受类别样本数量差异的影响。通过归一化处理，混淆矩阵中的数值转变为各个类别样本的百分比，即每行数值之和为 1。这样有助于更为便捷地比较不同类别间的性能差异，从而避免类别样本数量不同所带来的干扰。

（1）分别对字符型和单词型域名进行多分类检测

图 4.8 是 DGA 域名的多分类结果图。对角线上的深蓝色格子表示大多数实例被正确分类。例如，proslikefan、tinba、simda、emotet 等类别有高准确率，因为它们的大部分样本都被正确分类。对角线以外的深蓝色格子表示一种特定类别被常常误分类为另一种类别。例如，symmi、rovnix 和 conficker 主要被误分为被误分类为 gameover 和 ranbyus。一些类别之间存在混淆，可能由于它们具有相似的特征或模式，导致模型难以区分。比如，ramdo 和 ramnit 之间有一定的误分类。从两张图的分析中可以得出，基于字符和单词的分类模型都展现了很高的准确性和可靠性。

（a）字符级域名多分类



（b）单词级域名多分类

图 4.8 字符级域名和单词级域名多分类

对于字符级域名的分类检测，少数家族的部分域名被错分为其他家族，但是整体效果较为优异，大部分家族分类的准确率都可以达到 90%以上，甚至有些家族已经达到了 99%。以 proslikefan 和 qsnatch 为例，它们在自身类别的识别上准确率分别为 99.7%和 93.6%，显示出了较强的分类能力。然而，也存在一些误分类，如 ramnit 的误判率为 29.4%。这表明尽管模型总体上性能良好，但在某些特定类别之间仍然存在一定程度的混淆。

对于基于单词级域名的分类检测，模型仍然表现出较高的准确性，表明其对单词型域名具有出色的处理性能。例如，较难检测的 DGA 家族 emotete 和

simda 的检测准确率分别高达 99.9%和 99.8%，这一结果凸显了模型在识别这些类别的域名方面的精确度。这种准确性得益于模型在分词阶段采用了右移张量技术，有效减少了字符和词根词缀的干扰，降低了噪声。随后，利用 GRU-Attention 网络对单词的含义及其上下文关系进行学习，并生成了词向量，从而在识别过程中更快更准确地捕捉关键信息。

混淆矩阵可以用来识别模型在哪些类别上表现好，以及哪些类别可能需要进一步优化。它也可以帮助了解不同类别的误分类模式，从而改进分类算法。

(2)混合分类并与其他优秀模型比较

本节将验证整体模型的检测能力，并与多个优秀模型进行比较：

LSTM[53]：提出了一种 DGA 分类器，该分类器利用 LSTM 网络来预测 DGA 及其各自的家族，而无需先验特征提取。这是深度学习在这一领域的首次应用和深入分析，实现对 DGA 域名的有效检测。

Dilated-CNN[54]：该研究利用 CNN 进行 DGA 域名检测，旨在提升模型对域名的识别范围和准确性。研究团队引入了扩张的卷积核，还在前后扩张间隔中采用了残差连接技术，使模型能够更好地学习并保留域名序列中的重要特征，进一步提升识别准确率。

n-CBDC：此处模型与上面二分类实验所用模型以及参数完全相同。

表 4.17 显示了上面几种算法在多分类实验的精确率和 F1 值。值得注意的是多分类数据集选择的恶意域名家族种类很多，这些恶意域名家族有短域名，基于单词表的域名和普通的恶意域名，每个恶意域名家族的数量不同导致了精确率和 F1 值的波动。本文模型在大多数情况下，例如 ranbyus、qsnatch 和 gameover 这些类别的恶意域名，展现出了比 LSTM、Dilated-CNN 和 n-CBDC 更高的精确率。这意味着本文模型在预测恶意域名中，有更高准确率，减少了误报的情况。F1 分数是准确率和召回率的调和平均，反映了模型精确性和完整性的平衡。本文模型在 symmi、simda 和 murofet 等类别上，F1 分数显著高于其他三种模型，表明本文模型能更好地平衡精确性与完整性。

从表中数据可见，本文提出的模型在对各类别域名的检测中均展现了均衡且出色的性能，这凸显了模型的高度鲁棒性。本文模型的出色性能展示了其在实际应用中的巨大潜力，尤其适用于那些需要减少误报和漏报，以及对模型鲁棒性要求较高的场合。

表 4.17 多分类结果

域名类型	精确率（Precision）				F1-Measure（F1）			
	LSTM	Dilated-CNN	n-CBDC	本文模型	LSTM	Dilated-CNN	n-CBDC	本文模型
nymaim	0.9190	0.9764	0.9631	**0.9777**	0.9267	0.9563	0.9453	**0.9578**
symmi	0.6920	0.6944	0.6610	**0.7236**	0.7105	0.7478	0.7331	**0.7528**
virut	0.8290	0.8826	0.7869	**0.9128**	0.7397	0.7777	0.7837	**0.7658**
pykspa	0.9679	0.9692	0.9705	**0.9715**	0.9748	0.9806	0.9769	**0.9820**
gameover	0.9985	0.9985	0.9976	**0.9999**	0.9973	0.9976	0.9971	**0.9978**
ranbyus	0.7674	0.6799	0.7187	**0.8236**	0.7910	0.8064	0.8044	**0.8088**
murofet	0.8862	0.6981	0.7434	**0.8948**	0.7412	0.6644	0.6973	**0.7514**
ramnit	0.5822	0.6565	0.6086	**0.6896**	0.6354	0.6394	0.6323	**0.6416**
qsnatch	0.8470	0.9031	0.9122	**0.9328**	0.7458	0.8609	0.8187	**0.8768**
tinba	0.9536	0.9603	0.9602	**0.9550**	0.9744	0.9779	0.9771	**0.9763**
simda	0.9672	0.9609	0.9702	**0.9769**	0.9660	0.9790	0.9785	**0.9859**
banjori	0.9980	0.9995	0.9999	**0.9996**	0.9990	0.9996	0.9985	**0.9998**
rovnix	0.9996	0.9985	0.9984	**0.9996**	0.9977	0.9973	0.9970	**0.9976**
emotet	0.9186	0.9216	0.9236	**0.9346**	0.9568	0.9586	0.9521	**0.9586**
necurs	0.9507	0.9686	0.9623	**0.9722**	0.2353	0.2397	0.2278	**0.2328**

为了进一步分析分类的具体情况，制作了算法对不同类别分类结果归一化混淆矩阵，如图 4.9 所示。纵轴表示真实标签，横轴表示预测标签。

总结来说，在准确率、召回率和 F1 分数多个核心评价指标上，本文模型均展现出了比其他三个模型更优的性能。这些指标全面衡量了模型的性能，包括模型正确分类的能力、识别出所有正例的能力以及两者的综合表现。

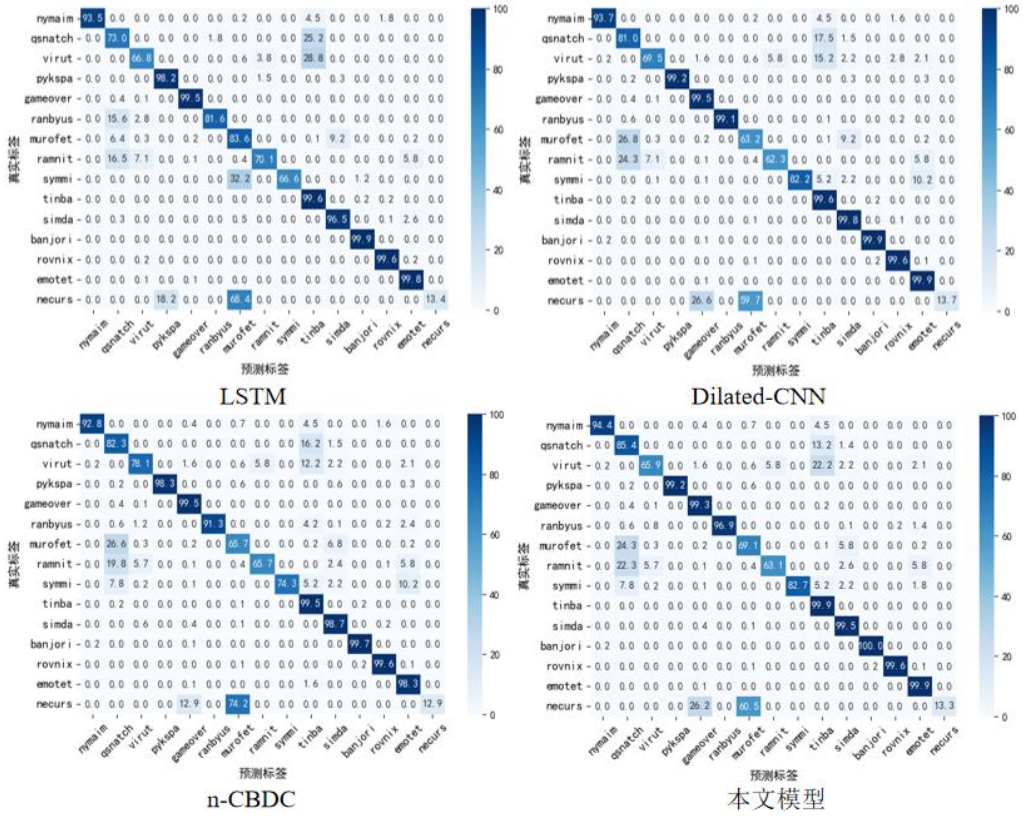


图 4.9 多分类结果的混淆矩阵

由于现实世界中的恶意域名种类繁多、变化莫测，因此一个具备均衡性能的模型能够更好地应对各种挑战。

4.3 本章小结

本章首先对第 2 章中提出的基于 n-gram 字典和 WGAN 的样本生成模型进行实验验证，发现生成域名避免了大量字符重复的情况，生成了与目标域名高度相似的生成域名。扩充后的数据集有效地提升了 DGA 域名检测分类模型的性能。然后对第 3 章中提出的异构 DGA 域名检测模型进行实验验证，该方法采用异构模型通过不同的特征提取方法，克服了对域名长度的敏感性，同时提高了对单词型域名的检测水平。从实验结果也可以发现，样本长度对模型的检测性能有影响。将难以检测的域名用特别的模型对其检测，提高 DGA 的整体检测水平。经过综合评估，验证了异构的 DGA 域名检测模型可以有效地提高 DGA 的检测性能，说明了模型的鲁棒性，在各种不同情况下都能保持高效的检测效果。

结论

本文聚焦于 DGA 域名检测领域，通过深度学习理论深入分析了当前该领域存在的一系列问题，并针对这些问题展开了 DGA 域名生成算法和检测算法的相关研究。由于 DGA 域名检测的域名数据相对容易获取，且仅需分析域名字符而不依赖其他信息，是现在僵尸网络检测的主流技术。近年来，研究人员对 DGA 域名分类方法进行了广泛的研究，基于深度学习的 DGA 域名分类器取得了显著的效果。然而，当前研究仍存在一些不足之处，有待进一步提升。

在此背景下，本文充分考察了当前主要的 DGA 域名检测与生成方法，并专注于解决 DGA 域名检测过程中的几个挑战。包括部分 DGA 域名家族样本稀缺性问题；短域名检测困难问题；基于单词的 DGA 域名与合法域名字符分布相似性问题。为应对这些问题，进行了深入研究并提出了相应的解决方案，取得了如下成果：

（1）在 DGA 域名生成方面，搭建的基于 n-gram 字典和 WGAN 的样本生成模型，通过判别网络和生成网络的对抗从真实域名中学习到 n-gram 的排列，然后根据排列顺序与 n-gram 字典建立索引，将其中的一些 n-gram 串接起来，从而合成域名。生成的域名与目标域名高度相似，可对小样本家族进行数据增强，扩充后的数据集有效地提升了 DGA 域名检测分类模型的性能。

（2）在 DGA 域名检测方面，本文针对现有的 DGA 域名检测方法对短域名和基于单词表的域名的特征提取不足、检测率不高的问题，构建了一种基于异构的 DGA 检测模型，该方法克服了对域名长度的敏感性。根据字符的长短，分为字符级域名检测模块和单词级域名检测模块。此模型充分考虑了短域名的特点和基于单词表域名的特点，采用不同的特征提取方式和处理方式。经过综合评估，验证了此模型可以有效地提高 DGA 的检测性能。克服了对样本长度的敏感性，对不同长度的样本具有稳定的检测能力。同时，针对基于单词表的 DGA 域名与合法域名字符分布相似性的问题，也取得了显著进展。相较于其他模型，本文构建的方法能够显著提升检测准确率，降低误报率，突显了本文提出模型检测的优越性能。

本文方法生成的恶意域名样本通过实验验证，显著增强了检测模型的数据信息。这些补充样本对于提高基于深度学习的域名检测模型的准确性、减少误识别率以及提升处理速度具有重要作用。然而，尽管本研究在恶意域名检测方

面显示了一定的价值，但仍存在一些不足之处，可作为未来研究的重点和方向。具体包括以下几个方面：

（1）在后续的实验阶段，检测模型在应对某些 DGA 域名家族时，其准确率尚未达到预期水平，尤其是一些不断变种和演化的 DGA 域名家族。鉴于目前的情况，需要进一步提升多分类检测的准确率，从而更有效地应对这些复杂多变的威胁。

（2）当前的研究虽然取得了一定的成果，但仅基于下载好的数据集进行模型训练与测试，尚不能完全反映模型在真实网络环境中的表现。为了全面评估算法模型的性能和准确性，未来的工作将专注于在真实网络环境。这样不仅能够检验模型在实时检测中的速度和效果，还能为模型的进一步优化提供宝贵的实践经验和数据支持，以更好地应对现实世界中不断变化的网络安全挑战。

参考文献

[1] Wu G, Wang X, Zhang J. PeerG: A P2P botnet detection method based on representation learning and graph contrastive learning[J]. Computers & Security, 2024: 103775.

[2] Shetu S F, Saifuzzaman M, Moon N N, et al. A survey of botnet in cyber security[C]. 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT). IEEE, 2019: 174-177.

[3] Tuan T A, Long H V, Taniar D. On detecting and classifying DGA botnets and their families[J]. Computers & Security, 2022, 113: 102549.

[4] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. Advances in neural information processing systems, 2014, 27.

[5] Tanaka F H K D S, Aranha C. Data augmentation using GANs[J]. arXiv preprint arXiv:1904.09135, 2019.

[6] Ren Z, Gao D, Zhu Y, et al. Generative adversarial networks driven by multi-domain information for improving the quality of generated samples in fault diagnosis[J]. Engineering Applications of Artificial Intelligence, 2023, 124: 106542.

[7] Klopries H, Schwung A. ITF-GAN: Synthetic time series dataset generation and manipulation by interpretable features[J]. Knowledge-Based Systems, 2024, 283: 111131.

[8] 傅伟, 钱丽萍, 朱晓慧. 基于改进 GAN 的恶意域名数据增强[J]. 计算机应用与软件, 2022, 39(03) : 308-315.

[9] Yu B, Pan J, Hu J, et al. Character level based detection of DGA domain names[C]. 2018 international joint conference on neural networks (IJCNN). IEEE, 2018: 1-8.

[10] Ren Y, Li H, Liu P, et al. CL-GAN: A GAN-based continual learning model for generating and detecting AGDs[J]. Computers & Security, 2023, 131: 103317.

[11] Huang J, Zhang G, Shen Y. DGA domain name detection based on SVM under grey wolf optimization algorithm[C]. 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS). IEEE, 2019: 245-248.

[12] Mu Z C. Predicting Domain Generation Algorithms with N-Gram Models[C]. 2022 International Conference on Big Data, Information and Computer Network (BDICN). IEEE, 2022: 31-38.

[13] Moubayed A, Injadat M N, Shami A, et al. Dns typo-squatting domain detection: A data analytics & machine learning based approach[C]. 2018 IEEE Global Communications Conference (GLOBECOM). IEEE, 2018: 1-7.

[14] Park K H, Song H M, Do Yoo J, et al. Unsupervised malicious domain detection with less labeling effort[J]. Computers & Security, 2022, 116: 102662.

[15] Tran D, Mac H, Tong V, et al. A LSTM based framework for handling multiclass imbalance in DGA botnet detection[J]. Neurocomputing, 2018, 275: 2401-2413.

[16] Curtin R R, Gardner A B, Grzonkowski S, et al. Detecting DGA domains with recurrent neural networks and side information[C]. Proceedings of the 14th international conference on availability, reliability and security. 2019: 1-10.

[17] Simran K, Balakrishna P, Vinayakumar R, et al. Deep learning based frameworks for handling imbalance in DGA, Email, and URL data analysis[C]. Computational Intelligence, Cyber Security and Computational Models. Models and Techniques for Intelligent Systems and Automation: 4th International Conference, ICC3 2019, Coimbatore, India, December 19–21, 2019, Revised Selected Papers 4. Springer Singapore, 2020: 93-104.

[18] Liu X, Liu J. DGA botnet detection method based on capsule network and k-means routing[J]. Neural Computing and Applications, 2022, 34(11): 8803-8821.

[19] Zhao D, Li H, Sun X, et al. Detecting DGA-based botnets through effective phonics-based features[J]. Future Generation Computer Systems, 2023, 143: 105-117.

[20] 王宇, 王祖朝, 潘瑞. 基于字符特征的 DGA 域名检测方法研究综述[J]. 计算机科学, 2023, 50(08) : 251-259.

[21] Hu X, Li M, Cheng G, et al. Towards Accurate DGA Detection based on Siamese Network with Insufficient Training Samples[C]. ICC 2022-IEEE International Conference on Communications. IEEE, 2022: 2670-2675.

[22] Fu Y, Yu L, Hambolu O, et al. Stealthy domain generation algorithms[J]. IEEE Transactions on Information Forensics and Security, 2017, 12(6): 1430-1443.

[23] García M, Maldonado S, Vairetti C. Efficient n-gram construction for text categorization using feature selection techniques[J]. Intelligent Data Analysis, 2021, 25(3): 509-525.

[24] Du K L, Swamy M N S. Neural networks and statistical learning[M]. Springer Science & Business Media, 2013.

[25] Hochreiter S, Schmidhuber J. LSTM can solve hard long time lag problems[J]. Advances in neural information processing systems, 1996, 9.

[26] Sibi P, Jones S A, Siddarth P. Analysis of different activation functions using back propagation neural networks[J]. Journal of theoretical and applied information technology, 2013, 47(3): 1264-1268.

[27] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks[C]. 2013 IEEE international conference on acoustics, speech and signal processing. Ieee, 2013: 6645-6649.

[28] Yang C, Lu T, Yan S, et al. N-trans: parallel detection algorithm for DGA domain names[J]. Future Internet, 2022, 14(7): 209.

[29] Yun X, Huang J, Wang Y, et al. Khaos: An adversarial neural network DGA with high anti-detection ability[J]. IEEE transactions on information forensics and security, 2019, 15: 2225-2240.

[30] Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks[C]. International conference on machine learning. PMLR, 2017: 214-223.

[31] Ahluwalia A. Impact study of length in detecting algorithmically generated domains[D]. 2018.

[32] Ahluwalia A, Traore I, Ganame K, et al. Detecting broad length algorithmically generated domains[C]. Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments: First International Conference, ISDDC 2017, Vancouver, BC, Canada, October 26-28, 2017, Proceedings 1. Springer International Publishing, 2017: 19-34.

[33] Schüppen S, Teubert D, Herrmann P, et al. {FANCI}: Feature-based automated {NXDomain} classification and intelligence[C]. 27th USENIX Security Symposium (USENIX Security 18). 2018: 1165-1181.

[34] Liang J, Chen S, Wei Z, et al. HAGDetector: Heterogeneous DGA domain name detection model[J]. Computers & Security, 2022, 120: 102803.

[35] Wang Q, Dong C, Jian S, et al. HANDOM: Heterogeneous attention network model for malicious domain detection[J]. Computers & Security, 2023, 125: 103059.

[36] 周飞燕，金林鹏，董军．卷积神经网络研究综述[J]．计算机学报，2017, 40(6):23.

[37] 胡金滨，唐旭清．人工神经网络的 BP 算法及其应用[J]．信息技术，2004, 28(4) : 1-4.

[38] 齐玉东，丁海强，司维超等．基于改进 CNN 的海军军事文本分类模型[J]．电光与控制，2020 , 27(05) : 68-73.

[39] 华帅，钟世立，李鑫鑫等．一种基于词嵌入模型和卷积神经网络的简化文本分类方法[J]．东莞理工学院学报，2022 , 29(05) : 69-78.

[40] Wendi D, Marwan N. Extended recurrence plot and quantification for noisy continuous dynamical systems[J]. Chaos: An Interdisciplinary Journal of Nonlinear Science, 2018, 28(8).

[41] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[42] Hu X, Chen H, Li M, et al. ReplaceDGA: BiLSTM based Adversarial DGA with High Anti-Detection Ability[J]. IEEE Transactions on Information Forensics and Security, 2023.

[43] Wang H, Tang Z, Li H, et al. CI_GRU: An efficient DGA botnet classification model based on an attention recurrence plot[J]. Computer Networks, 2023, 235: 109992.

[44] Luo P, Ren J, Peng Z, et al. Differentiable learning-to-normalize via switchable normalization[J]. arXiv preprint arXiv:1806.10779, 2018.

[45] Ba J L, Kiros J R, Hinton G E. Layer normalization[J]. arXiv preprint arXiv:1607.06450, 2016.

[46] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]. International conference on machine learning. pmlr, 2015: 448-456.

[47] Ulyanov D, Vedaldi A, Lempitsky V. Instance normalization: The missing ingredient for fast stylization[J]. arXiv preprint arXiv:1607.08022, 2016.

[48] Corley I, Lwowski J, Hoffman J. Domaingan: generating adversarial examples to attack domain generation algorithm classifiers[J]. arXiv preprint arXiv:1911.06285, 2019.

[49] Anderson H S, Woodbridge J, Filar B. DeepDGA: Adversarially-tuned domain generation and detection[C]. Proceedings of the 2016 ACM workshop on artificial intelligence and security. 2016: 13-21.

[50] Xu C, Shen J, Du X. Detection method of domain names generated by DGAs based on semantic representation and deep neural network[J]. Computers & Security, 2019, 85: 77-88.

[51] Vranken H, Alizadeh H. Detection of DGA-generated domain names with TF-IDF[J]. Electronics, 2022, 11(3): 414.

[52] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

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

[54] Zhou S, Lin L, Yuan J, et al. CNN-based DGA detection with high coverage[C]. 2019 IEEE international conference on intelligence and security informatics (ISI). IEEE, 2019: 62-67.