兰州理工大学

硕士论文

学校代号 10731　　学　　号 202085400123

分 类 号 TP391　　密　　级 公　开

兰州理工大学

LANZHOU UNIVERSITY OF TECHNOLOGY

专业学位硕士学位论文

基于域名特征融合的恶意域名检测方法研究

学位申请人姓名 申宋彦

培　养　单　位 计算机与通信学院

导师姓名及职称 赵宏 教授

学　科　专　业 计算机技术

研　究　方　向 模式识别与人工智能

论文提交日期 2023 年 3 月 20 日

学校代号：10731

学　　号：202085400123

密　　级：公　开

兰州理工大学专业学位硕士学位论文

基于域名特征融合的恶意域名检测方法研究

学位申请人姓名：　申宋彦

导师姓名及职称：　赵宏 教授

培　养　单　位：　计算机与通信学院

专　业　名　称：　计算机技术

论文提交日期：　2023 年 3 月 20 日

论文答辩日期：　2023 年 5 月 25 日

答辩委员会主席：　王志伟 高级工程师

Research on Malicious Domain Name Detection Method Based on Domain Name Feature Fusion

by

SHEN Songyan

B.E. (Yuncheng University) 2019

A thesis submitted in partial satisfaction of the

Requirements for the degree of

Master of Professional

in

Computer Technology

in the

School of Computer and Communication

Lanzhou University of Technology

Supervisor

Professor ZHAO Hong

May, 2023

目 录

摘　要

域名系统（Domain Name System，DNS）是一个分布式数据库系统，用于实现域名与 IP（Internet Protocol）地址之间的相互映射。DNS 允许互联网用户通过使用易于记忆的域名代替 IP 地址来访问网站。然而，DNS 本身并不具备防御功能，使得 DNS 成为攻击者攻击网络的入口。例如 DNS 劫持和欺骗、网络钓鱼、恶意软件传播和网络诈骗等，这些攻击给网络和用户的安全带来了重大的影响。因此，设计分类模型快速准确地检测出恶意域名、防止域名被恶意用户滥用是非常重要的，可以有效减少网络的安全威胁。

本文旨在综合考虑域名检测的精度、范围、实际处理时间以及模型大小等因素，利用深度学习的卷积神经网络和自然语言处理相关理论和技术，评估待检测的域名。本文提出了两种新型的域名检测模型，包括基于字符特征和词特征融合的恶意域名检测模型 CWNet（Character and Word Network）和基于全卷积的快速恶意域名检测模型 LW-CWNet（Light-Weighted Character and Word Network）。主要研究工作如下：

1.针对现有恶意域名检测方法对域名生成算法(Domain Generation Algorithm，DGA)随机产生的恶意域名检测性能不高，且对由随机单词组成的恶意域名检测效果较差的问题，提出一种基于字符和词特征融合的恶意域名检测模型 CWNet。该模型首先利用并行卷积神经网络分别提取域名中字符和词的特征，然后拼接提取到的两种特征，构造成含域名字符和词信息的融合特征，最后输出层使用 Softmax 函数得到待测域名的分类预测结果。损失函数使用聚焦损失函数，减少简单样本对损失函数的影响，更加关注难分类的样本，提高模型分类性能。实验结果表明，该模型可以提高对恶意域名的检测性能，对难度较大的恶意域名家族的检测能力有明显提升。

2.针对 CWNet 模型计算量和参数数量过多，模型占用内存较大，难以在现实场景中得到应用的问题，在 CWNet 模型基础之上，提出一种基于全卷积的快速恶意域名检测模型 LW-CWNet。该模型首先使用深度可分离卷积代替传统的卷积神经网络模型构建轻量级的并行卷积神经网络，然后使用参数量较少的全局平均池化层代替全连接层，得到轻量化恶意域名检测模型，最后采用标签平滑方法防止模型在训练过程中过拟合，提高模型的泛化能力。实验结果表明，LW-CWNet 模型可以在参数数量显著减少的情况下，保持较高的域名检测精度。与其他恶意域名检测模型相比，该模型在对域名分类的效率、精度和模型大小等方面有显著提高。

关键词：恶意域名；卷积神经网络；特征融合；深度可分离卷积；全局平均池化

Abstract

The Domain Name System (DNS) is a distributed database system used to map domain names to each other and to IP (Internet Protocol) addresses. DNS allows users on the Internet to access websites by using easy-to-remember domain names instead of IP addresses. However, DNS itself is not defensible, making it an entry point for attackers to attack the network. Examples include DNS hijacking and spoofing, phishing, malware distribution, and cyber fraud, and these attacks have significant implications for the security of networks and users. Therefore, it is important to design classification models to quickly and accurately detect malicious domain names and prevent them from being abused by malicious users, so that the security threats to the network can be effectively reduced.

This article aims to evaluate the domain names to be detected by considering the accuracy, range, actual processing time and model size of domain name detection, using deep learning convolutional neural networks and theories and techniques related to natural language processing. In this article, two novel detection models are proposed, including CWNet (Character and Word Network), a malicious domain name detection model based on the fusion of character and word features, and LW-CWNet (Light-Weighted Character and Word Network), a fast malicious domain name detection model based on full convolution network. The main research work of this article is as follows:

1.In order to solve the problem that the existing malicious domain name detection methods do not have high performance in detecting malicious domain names randomly generated by Domain Generation Algorithm (DGA) and have poor results in detecting malicious domain names composed of random words, we propose a malicious domain name detection model CWNet based on the fusion of character and word features. The model first uses parallel convolutional neural networks to extract the character and word features of domain names separately, then splices the two extracted features to construct a fusion feature containing the character and word information of domain names, and finally the output layer uses Softmax function to obtain the classification prediction results of domain names to be tested. The loss function uses Focal Loss to reduce the effect of simple samples on the loss function, focus more on difficult to classify samples, and improve the model classification performance. The experimental results show that the model can improve the detection performance of malicious domain names, and the detection ability of more

challenging malicious domain name families is improved more significantly.

2.To address the problem that the CWNet model has too many computations and parameters, and the model occupies a large amount of memory, which makes it difficult to be applied in real scenarios, a fast malicious domain name detection model based on full convolution LW-CWNet is proposed on top of the CWNet model.This model first constructs a lightweight parallel convolutional neural network by using deep separable convolution instead of the traditional convolutional The model firstly constructs a lightweight parallel convolutional neural network by using deep separable convolution instead of the traditional convolutional neural network model, then uses a global average pooling layer with fewer parameters instead of a fully connected layer to obtain a lightweight malicious domain name detection model, and finally uses label smoothing to prevent the model from overfitting during the training process and improve the generalization ability of the model. The experimental results show that the LW-CWNet model can maintain high domain name detection accuracy with a significantly reduced number of parameters. Compared with other malicious domain name detection models, the model has significantly improved the efficiency, accuracy and model size in classifying domain names.

Keywords: Malicious domain name; Convolution neural network; Feature fusion; Depth separable convolution; Global average pooling

插图索引

附表索引

第1章 绪 论

1.1 课题的研究背景与意义

在互联网上，每台计算机和其他通信设备相互通信时都需要一个 IP 地址。以 IPv4 地址为例，IP 地址是一个 32 位的二进制数，通常用点分十进制表示成 a.b.c.d 的形式。由于 IP 地址是由数字和分隔符组成的，不易记忆。因此，可以通过域名来代替数字型的 IP 地址，每个域名都与特定的 IP 地址对应。这种映射使用户方便地访问对应的主机。例如，淘宝网站的 IP 地址是：210.75.225.254，通过字符化的 www.taobao.com 可以实现映射。域名系统是一个分布式数据库系统，用于实现域名与 IP 地址之间的相互映射，域名解析成 IP 地址的工作由域名系统服务器完成。然而，在早期设计中，域名系统并没有考虑到网络安全相关的问题，存在许多安全性漏洞。其中之一是缓冲区溢出漏洞，攻击者可以在域名系统上运行各种指令。另一个是拒绝服务漏洞，攻击后的域名系统服务器将不能提供正常服务，使得其所管辖的子网无法正常工作。此外，域名系统面临分布式拒绝服务攻击、区域信息泄露、缓存投毒等威胁。因此，为了应对网络攻击，建立高效的网络安全环境是十分必要的[1]。

随着互联网的发展，网络安全面临的风险和影响越来越严峻。恶意域名是网络安全的一个关键问题，可以用来进行欺骗、传播病毒或窃取敏感信息，对互联网和个人的信息安全造成严重危害。为了保护网络安全，许多机构和企业已经采用了各种方法来检测和防范利用恶意域名的各种攻击手段。2022 年国家互联网应急中心发布的监测数据中，我国互联网网络安全环境与恶意域名有关的情况如下：境内感染主机排名前五位的木马或僵尸网络家族主要是 Ramnit、Chacha、Mirai、Blackmoon、Floxif，这五种木马或僵尸网络感染主机数量约占全部感染主机总数的 88.2%，其中 Ramnit、Mirai 这两个 DGA（Domain Generation Algorithm）家族是由域名生成算法随机产生，相对于传统硬编码的恶意域名危害更大。境内被篡改网站数量为 3713 个，被植入后门的网站数量为 1960 个，网站的仿冒页面有 7740 个，涉及域名 7624 个，IP 地址 403 个。网络空间中出现的恶意域名攻击越来越多，为了应对恶意域名的威胁，需要提高恶意域名检测系统的检测速度和准确性。因此，如何更快速准确地识别和防范恶意域名已成为网络安全领域研究的一个重要方向。

由于科学技术的不断发展，恶意域名检测技术也更加多样。从最初的基于规则的方法到现在的基于深度学习的方法，恶意域名检测技术已经取得了显著的进展。然而，攻击者的技术变得越来越复杂以及大量不同类型的恶意域名出现，恶

意域名检测仍然是一个具有挑战性的问题。因此，需要不断改进和更新恶意域名检测方法，以应对不断变化的威胁网络安全的环境[2]。

恶意域名检测研究意义重大。首先，恶意域名可能被用于欺骗、窃取敏感信息等不良行为，故准确识别和防范恶意域名有助于保护用户数据和隐私。其次，实时准确的恶意域名检测可防范各类恶意域名攻击，降低网络安全威胁，提高网络安全水平。最后，恶意域名检测是网络安全领域必须关注的重要方向，研究和改进检测技术可推进技术进步、为网络安全研究和开发提供科学依据[3]。

1.2 国内外研究现状

恶意域名检测是一个复杂且持续发展的问题。目前，有许多不同的研究方法和技术用于检测恶意域名，现有恶意域名检测模型主要包括基于域名黑名单规则和特征的方法、基于机器学习的域名特征组合过滤、基于深度学习算法检测三大类[4, 5]，如图 1.1 所示：

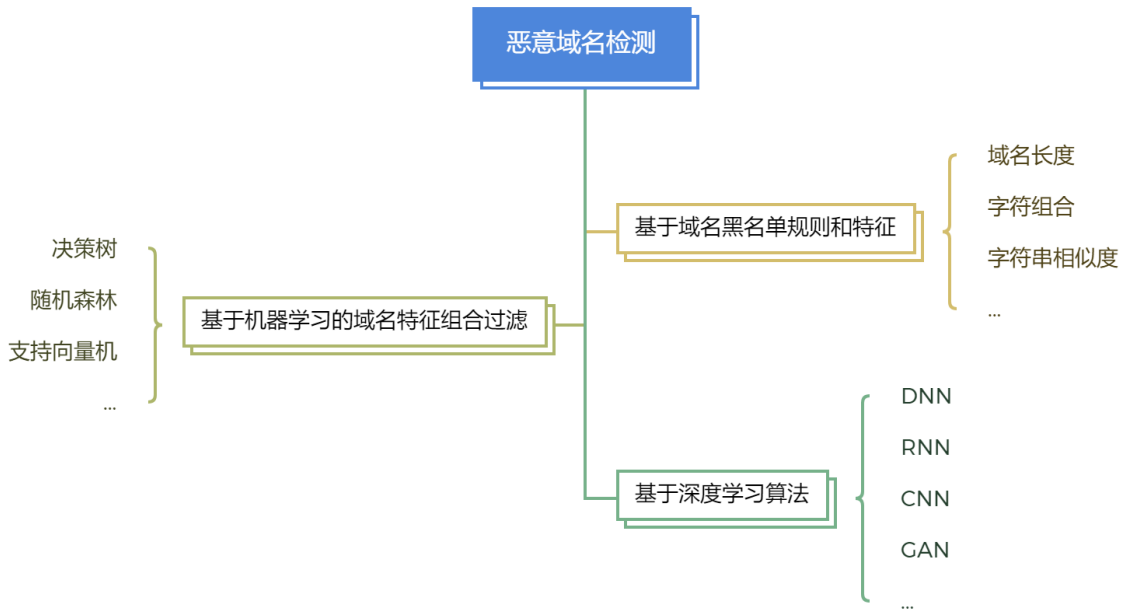

图 1.1 国内外研究现状图

（1）基于域名黑名单规则和特征的方法

基于域名黑名单规则和特征的恶意域名检测方法通过分析域名黑名单中的特征，如字符长度、字符种类和流量信息等，用于识别恶意域名。这种方法通常需要手动定义规则，以识别域名中可疑的特征。域名长度：过长或过短的域名可能是恶意的。字符串相似度：如果两个域名相似，则可能是恶意的。TTL 长短：特定范围内的 TTL 的使用比例，TTL 设置较低的则可能是恶意域名。域名和 IP 映射多对多关系：IP 对应的不同国家数，目标域名的 IP 共享域名数较多则可能是恶意域名。赵宏等[6]提出了一种基于词法特征的恶意域名快速检测算法，根据待测域名与黑名单域名之间的编辑距离值快速判断是否为恶意域名，旨在解决现有

域名检测方法实时性不强的问题，结果表明在准确率和检测速率方面表现较好。Hong 等[7]使用基于 N-Gram 的恶意域名检测算法，将合法域名建立子字符串集合，根据在子字符串集合中出现的次数计算每个子字符串的权重值，将待测域名进行 N-Gram 方法的分析，基于其子字符串的权重值计算出其信誉值，并通过设置阈值进行域名的分类。Zhao 等[8]提出了一种两阶段的恶意域名检测算法，第一阶段使用黑名单与待测域名进行比对得到初步预测结果，第二阶段则利用 N-Gram 模型计算待测域名的声誉值，并根据其声誉值来判断待测域名是否为恶意域名，该方法可以有效地检测恶意域名。Saiyan 等[9]通过与域名黑名单字符匹配的方式，快速给出待测域名的判断。这些方法旨在通过将恶意域名与已知恶意域名的数据库进行匹配，或通过检查域名的各种特征（如长度、结构或内容）来识别恶意域名。这些方法过于依赖域名黑名单，可能无法检测新的或高级形式的恶意域名，导致类 DGA 算法新生成的恶意域名无法被及时检测并过滤。

（2）基于机器学习域名特征组合过滤的方法

基于机器学习的恶意域名检测方法是一种自动识别恶意域名的方法，这种方法通常使用机器学习算法，如 J48 决策树，随机森林，支持向量机，朴素贝叶斯等，对域名的多个特征进行分析，以识别恶意域名[10, 11]。马栋林等[12]提出了一种基于 Rf-C5 算法的恶意域名检测算法，解决了字符特征数量过多造成计算开销大的问题，减少了域名检测的时间。Yang 等[13]针对基于单词的 DGA，从词频、词性和词相关性等方面提取了 16 维特征，使用随机森林进行域名检测，效果较好。Cucchiarelli 等[14]通过比较待测域名中的二元组和三元组与已知恶意或良性域名中的二元组和三元组的相似度，使用 Kullback-Leibner 散度和 Jaccard 指数评估待测域名，结果表明该方法仅利用域名的词汇特征有良好的准确性。Alshdadi 等[15]使用个体、分布式和混合特征的机器学习模型，显著改善了现有的恶意域名机器学习技术的性能，相比 C4.5、AdaBoost 和 LogitBoost，支持向量机分类器有更高的准确率。Cheng 等[16]提出了一种基于 AdaBoost 的轻量级检测方法，旨在通过探索异常的 WHOIS 记录来主动检测恶意域名，以提高域名注册的安全性，用于识别恶意域名，在大规模数据集上进行的实验表明，所提出的方法取得了较好的结果。张斌等[17]提出一种新的恶意域名检测方法，将域名解析记录表示为异质信息网络中的节点和边，采用基于元路径的广度优先网络遍历算法提取关联解析信息，引入请求时间来刻画域名之间的相关性，提高检测样本覆盖率，通过域名特征向量之间的欧氏距离量化域名之间的关联性，进而构建有监督分类器进行恶意域名检测。国内外的研究现状显示，基于机器学习的恶意域名检测方法在准确性和效率方面有了很大的提高，但是仍存在一些挑战[18]。利用机器学习方法可以有效提高恶意域名的检测性能，但是域名特征需要人工设计，无法满足恶意域名检测的实时性要求[19-21]。

（3）基于深度学习的恶意域名检测方法

近年来，国内外学者基于深度学习的方法，对恶意域名进行了大量的检测研究。研究方法主要有基于深度神经网络（DNN）、卷积神经网络（CNN）、循环神经网络（RNN）、生成对抗网络（GAN）等[22-25]。这些方法利用大量的恶意域名和合法域名数据集进行训练，以提高识别恶意域名的准确率[26-28]。目前，基于深度学习的方法在恶意域名检测方面取得了显著的成果，并被广泛应用于实际场景。然而，由于攻击者技术不断提高，以及大量新域名的出现，恶意域名检测仍然是一个持续的挑战。因此，需要不断改进和更新恶意域名检测方法，以应对不断变化的威胁环境[29]。国内外的研究表明，基于深度学习的恶意域名检测方法具有很高的准确率和可靠性，已经成为网络安全领域的重要研究方向[30, 31]。

深度学习方法可以自动提取域名的深度特征，逐渐成为恶意域名检测的主流方法[32, 33]。Woodbridge 等[34]提出了提出一种基于 LSTM 网络的 DGA 分类模型，无需预先提取特征，可预测 DGA 及其家族，具有较高的准确率和低误报率。Yu 等[35, 36]提出一种基于并行 CNN 的模型，可以得到域名中不同长度的 N-Gram 信息，成功应用在域名、URL 等分类任务中。杨路辉等[37]在文献[36]的基础上对模型结构进行了改进，增加了卷积分支用于提取域名的深层字符特征，之后对得到的不同的卷积特征进行融合，域名信息更大程度被利用，提高对 DGA 域名的检测率。王志强等[38]提出使用 DCNN 的方法进行 DGA 域名检测，域名的深层特征可以通过动态调整参数更大范围被利用，从而有效识别 DGA 域名。李晓东等[39]提出对域名的字符特征、词根特征、分词特征进行融合，得到域名信息更加完整的融合向量，之后域名特征使用并行的卷积核不同的一维卷积神经网络提取，然后作为双向循环门控网络的输入挖掘在时序上的更深层特征，实现域名分类检测[40, 41]。Yang 等[42]提出了一种实时恶意域名检测系统 fast3DS，该系统采用了轻量级的全卷积检测模型以及多种高效的数据获取、过滤和推理方案，使其能够快速检测恶意域名，该检测方案可以在参数减少的情况下有较好准确性，且系统处理能力较传统检测系统显著提高。Aarthi 等[43]提出了一种基于循环神经网络算法识别潜在恶意域名的方法，将 URL 分成子域、域和域后缀，基于此训练神经网络以将给定的数据集分类为恶意或良性，该方法在测试集上具有较高的准确性。Namgung 等[44]提出一种基于双向长短期记忆神经网络域名检测方法，该方法采用 BiLSTM 学习双向信息，提高检测性能，并通过 Attention 机制从域名序列中学习局部和全局信息，结果表明可实现更准确的 DGA 域名检测。Singh 等[45]提出使用多层卷积神经网络进行恶意 URL 检测，结果表明当模型使用多层 CNN 时，可以提高恶意检测准确率。Solovyeva 等[46]提出一种基于可分离卷积的网络结构，包括嵌入层、可分离卷积层、卷积层和全局平均池化层，用于文本二分类和多分类。该网络旨在获得高准确率的同时减少模型复杂度。Huang 等[47]提出了一种基于

TextCNN 的恶意域名检测模型，该模型能够结合不同的网络模型的优势，在恶意域名检测中有更好的检测效果。Yadav 等[48]提出一种轻量级文本分类模型可用于恶意域名检测，模型可训练参数更少，减小了内存消耗，提高了检测效率。

1.3 主要研究内容

本文提出了基于字符特征和词特征融合的恶意域名检测模型和基于全卷积的快速恶意域名检测模型，主要研究内容如下：

（1）基于字符特征和词特征融合的恶意域名检测模型

针对现有恶意域名检测方法对于域名生成算法随机产生的恶意域名检测性能不高、以及对由随机单词组成的恶意域名检测效果较差的问题，提出了一种基于字符和词特征融合的恶意域名检测算法 CWNet（Character and Word Network）。该模型首先对待测域名使用 Word2Vec 进行向量化得到域名的向量表示，之后使用并行卷积神经网络利用多个不同尺寸大小的卷积核提取域名字符级特征和词级特征，然后将这两种特征进行拼接，构造成融合特征。最后，输出层使用 Softmax 函数得到待测域名合法或恶意的分类结果。实验结果表明，该算法可以提升对恶意域名的检测能力，并对更具挑战性的恶意域名家族的检测准确率提升效果更为明显。

（2）基于全卷积的快速恶意域名检测模型

针对 CWNet 模型计算量和参数量过多，难以在现实场景中得到应用，本节在提出的 CWNet 模型基础之上，提出了一种轻量级全卷积的快速恶意域名检测模型 LW-CWNet（Light-Weighted Character and Word Network）。首先通过使用深度可分离卷积代替传统的卷积神经网络模型构建轻量化并行卷积神经网络 DS-CWNet（Depthwise Seperable-Character and Word Network）；之后使用轻量级全局平均池化（Lightweight Global Average Pooling，LGAP）代替全连接层，减少训练参数数量；最后采用标签平滑（Label Smoothing，LS）方法修正损失函数防止模型在训练过程中过拟合，提高模型的泛化能力。实验部分评估了深度可分离卷积，全局平均池和标签平滑对模型性能和存储空间的影响。结果表明，所提出的 LW-CWNet 模型可以在参数显著减少的同时保持高精度的恶意域名检测。与其他恶意域名检测模型相比，在对域名分类的效率，精度和模型的大小上有显著提升。

1.4 论文组织结构

本论文针对现有研究存在的问题和不足，提出了两种恶意域名检测模型，达到了预期效果。论文共分四个章节，结构安排如图 1.2 所示：

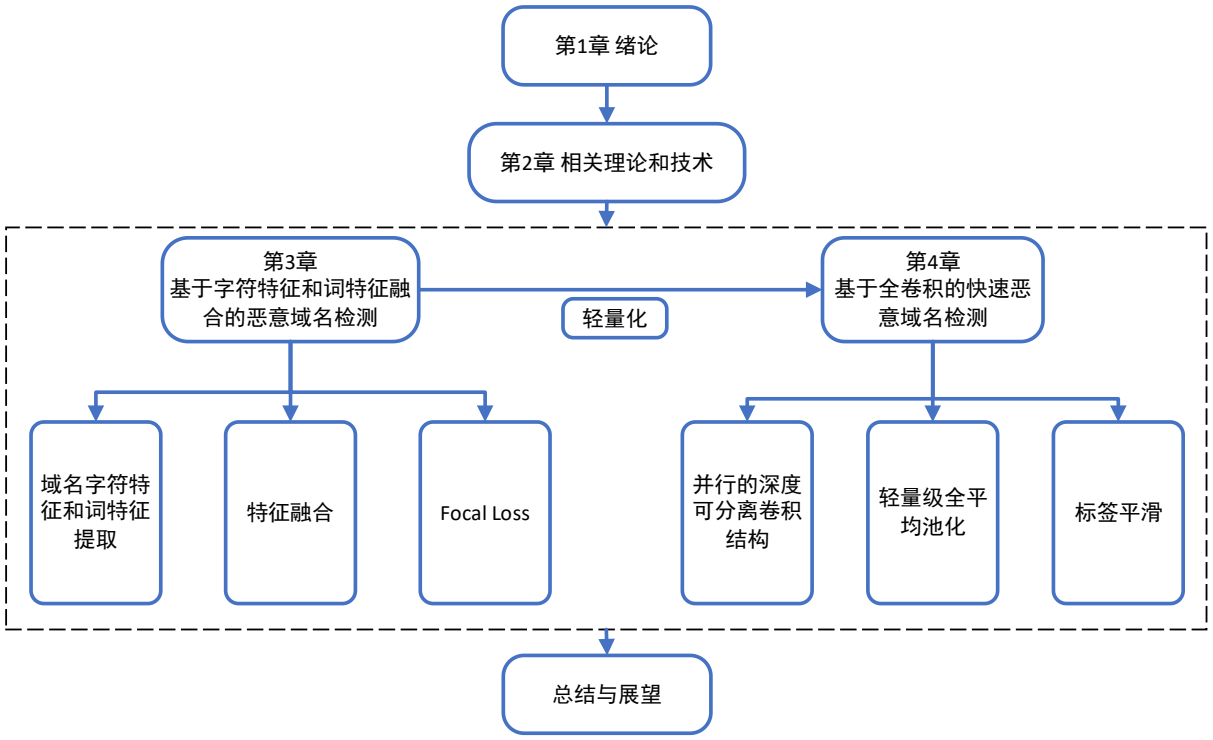

图 1.2 论文章节结构图

第 1 章 绪论。详细介绍了本文课题的研究背景与意义，之后阐述了恶意域名检测国内外研究现状，最后介绍本论文的研究内容和组织结构。

第 2 章 相关理论和技术。介绍了域名的相关知识、向量化方法、不同的神经网络模型，阐述了不同类型的损失函数和域名分类的主要性能评价指标。

第 3 章 基于字符特征和词特征融合的恶意域名检测。首先介绍了 CWNet 的网络结构，之后对各个模块进行详细阐述，最后对本章模型进行了详细的实验，验证了该模型有很强的检测能力，能有效检测各种类型的恶意域名。

第 4 章 基于全卷积的快速恶意域名检测。在第 3 章提出的 CWNet 基础之上对模型进行了改进，介绍了改进后的模型 LW-CWNet 的网络结构，对各个改进方法进行了详细阐述，通过进行大量的实验，验证了该模型具有更好的实际应用价值，在现实场景中能够更加高效地进行恶意域名检测。

本文的最后是总结与展望，对本文的主要研究内容和研究成果进行了重点归纳和总结，提出了目前研究内容可供改进的方向以及对未来研究工作的展望。

第2章 相关理论和技术

2.1 域名

域名是一种用于标识互联网上计算机、服务器等网络资源的字符序列，通常由多个由点分隔的部分组成，比如"xinhuanet.com"。通过域名系统将域名解析成相应的 IP 地址，从而让人们可以通过易于记忆的名称来访问特定的网络资源。域名的定义可以简单概括为：域名是用于唯一标识互联网上特定网络资源的字符序列。域名在互联网上扮演着非常重要的角色，是人们访问互联网上特定网络资源的重要途径之一。

2.1.1 域名系统

域名系统（Domain Name System，DNS）是互联网中将域名与 IP 地址相互映射的一种机制，可以将人类易于记忆的域名转换为计算机可识别的 IP 地址，以便于网络通信和数据传输。DNS 是互联网中不可或缺的基础设施之一。DNS 是一种分布式的命名系统，可以将域名分配给不同的 IP 地址，以实现网站、电子邮件服务器、FTP 服务器等互联网服务的定位。DNS 的运行过程中，客户端向本地 DNS 服务器发送域名请求，如果本地 DNS 服务器没有缓存该域名对应的 IP 地址，会向更高级别的 DNS 服务器发送请求，直到最终找到域名对应的 IP 地址并将结果返回给客户端。DNS 系统由多级 DNS 服务器组成，通过协议和算法协同工作，使得整个系统能够高效地运转。DNS 系统具有高可用性、可扩展性和灵活性等特点，可以根据需要对 DNS 服务器进行添加、删除和修改。同时，DNS 还支持多种类型的记录，例如 A 记录、MX 记录、CNAME 记录等，使得管理员可以更灵活地配置 DNS 服务器，满足不同的需求。DNS 还支持缓存机制，可以提高域名解析的速度和效率，并减轻 DNS 服务器的负担。DNS 的安全性也很重要，因为 DNS 缓存污染和 DNS 劫持等攻击会对网络安全产生威胁，因此 DNS 服务器需要采取相应的安全措施来保护自身和用户的安全。为了提高 DNS 的效率和可靠性，现代 DNS 系统引入了多种技术，例如任播（Anycast）、DNS 安全扩展（DNSSEC）、扩展 DNS（EDNS）等。任播技术可以将多个 DNS 服务器部署在不同的地理位置上，使得用户能够就近访问最快速的 DNS 服务器，从而提高 DNS 的响应速度和可用性。DNSSEC 则可以对 DNS 响应进行数字签名，确保用户接收到的域名解析结果是真实可信的。EDNS 则支持更大的 UDP 数据包和 TCP 协议，以提高 DNS 的扩展性和兼容性。

DNS 系统是互联网中极为重要的基础设施之一，使得用户能够轻松地访问互

联网上的各种服务，并且具有高可用性和灵活性等优点。随着互联网的不断发展，DNS 系统也在不断改进和完善，以应对日益增长的用户需求和安全挑战。

2.1.2 域名解析过程

域名解析是将域名解析为相应 IP 地址，以便在互联网上找到正确的服务器并访问网站。域名解析通常由域名系统完成，如图 2.1 所示是域名解析的基本过程：先检查浏览器缓存和配置文件，如果没有相关信息，浏览器向本地 DNS 服务器发送域名解析请求。如果本地 DNS 服务器缓存了该域名的 IP 地址，则返回 IP 地址给浏览器；否则，本地 DNS 服务器向根 DNS 服务器发送请求。根 DNS 服务器返回包含顶级域名服务器的 IP 地址列表的响应。如请求的域名为".com"，则返回包含".com"顶级域名服务器的 IP 地址列表。本地 DNS 服务器向".com"顶级域名服务器发送请求。".com"顶级域名服务器返回包含二级域名服务器的 IP 地址列表的响应。如请求的域名为"xinhuanet.com"，则返回包含"xinhuanet.com"二级域名服务器的 IP 地址列表。本地 DNS 服务器向" xinhuanet.com"二级域名服务器发送请求。" xinhuanet.com"二级域名服务器返回包含请求的域名的 IP 地址的响应。本地 DNS 服务器将该 IP 地址返回给浏览器，并将其缓存以备下一次请求使用。浏览器使用该 IP 地址向服务器发起请求，以获取网站的内容。

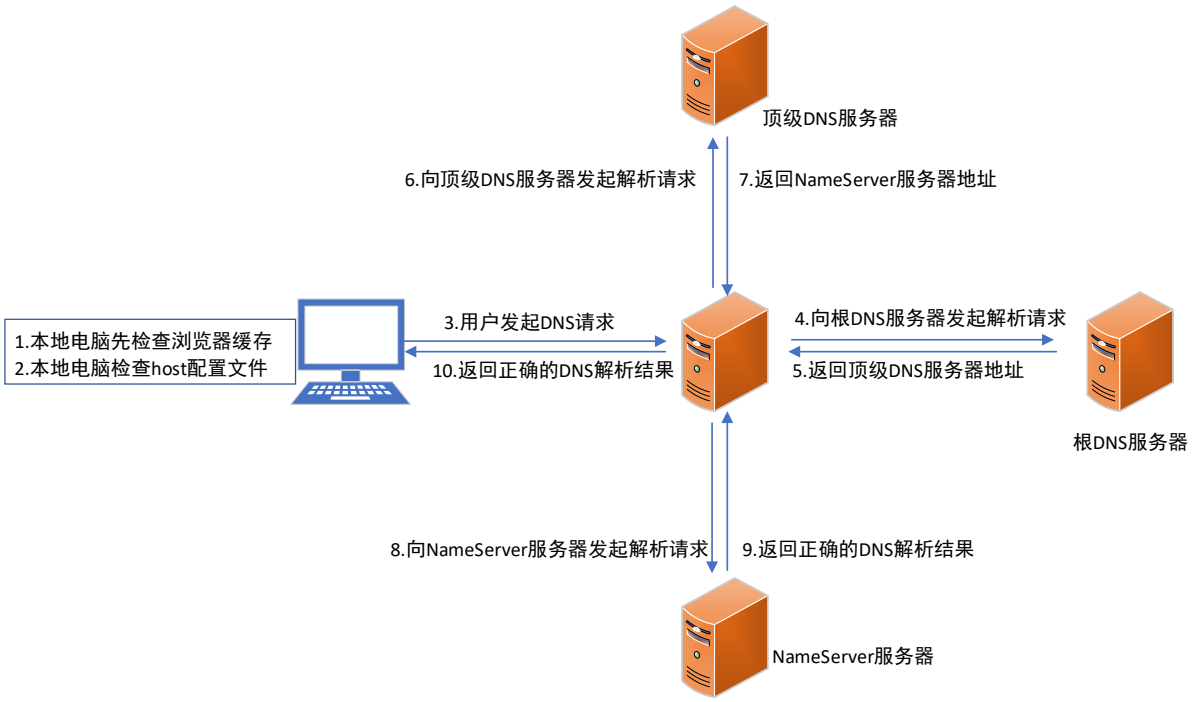

图 2.1 域名解析流程图

2.1.3 恶意域名

恶意域名是指用于进行网络攻击、欺诈、传播恶意软件等非法行为的域名。这些域名可能被用于欺骗用户，窃取个人信息，传播恶意软件等。恶意域名的种

类比较多，其中一些常见的包括：

（1）仿冒域名（Phishing domains）：仿冒域名是一种欺诈行为，攻击者通过注册与受害者信任的机构或网站相似的域名，欺骗用户输入个人信息，从而达到窃取信息的目的。

（2）命名混淆域名（Typosquatting domains）：攻击者会注册与受害者常用的域名相似，但拼写错误的域名，希望用户误输入域名，进而访问到恶意站点[49]。

（3）垃圾邮件域名（Spam domains）：攻击者会使用大量的垃圾邮件域名来发送垃圾邮件，传播恶意软件等。

（4）恶意软件域名（Malware domains）：攻击者会在恶意软件中插入用于连接恶意服务器的域名，从而将恶意软件传播到受害者的设备中。

恶意域名的危害包括：窃取用户的个人信息，感染用户设备，导致设备的性能下降或崩溃，传播恶意软件，攻击其他设备，甚至造成数据泄露，金融损失等严重后果。因此，及时识别和阻止恶意域名是非常重要的[50]。

2.1.4 域名生成算法

域名生成算法（DGA）是一种用于动态生成恶意域名的算法。攻击者使用DGA 算法生成一组或多组域名，然后将这些域名用于控制恶意软件、攻击其他计算机等非法行为。DGA 的作用是在避开安全设备的检测的同时，使恶意软件能够快速、有效地连接到攻击者控制的服务器，从而执行各种攻击行为，如窃取个人信息、勒索软件、挖矿、DDoS 攻击等。DGA 的危害包括：使恶意软件难以被检测和清除，增加了恶意软件对目标设备的操控能力，以及绕过防御措施，从而危害到个人隐私和企业信息安全等。因此，及时检测和阻止 DGA 生成的恶意域名非常重要[8]。

DGA 是一种恶意软件技术，用于生成大量随机的域名，以避开网络安全系统的检测。根据种子性质，DGA 可以分为以下几类：

（1）时间戳种子：使用当前时间作为种子，生成域名。这种 DGA 通常使用间隔较短的时间戳，如每秒钟或每分钟生成一次，以确保生成的域名数量足够多，同时避免生成相同的域名。

（2）伪随机数种子：使用伪随机数生成器生成种子，再根据种子生成域名。这种 DGA 使用的伪随机数生成器通常是基于某些算法的，如线性同余生成器或多项式生成器。由于伪随机数是根据种子生成的，因此如果攻击者掌握了生成域名的算法和种子，就可以预测未来生成的域名。

（3）真随机数种子：使用真随机数生成器生成种子，再根据种子生成域名。这种 DGA 使用的真随机数生成器通常是基于硬件的，如基于放射性衰变或热噪声的真随机数生成器。由于真随机数是无法预测的，因此这种 DGA 的域名生成更加随机。

（4）外部数据种子：使用外部数据作为种子，如网络流量、天气数据、新闻事件等。这种 DGA 的域名生成会受到外部数据的影响，因此更加难以预测和检测。

又可根据生成算法的不同分为以下几类：

（1）字典算法：使用预先定义的单词或字符组合作为基础字典，再根据一定的规则进行变换和组合，生成域名。这种 DGA 的域名通常是有意义的，可能与攻击主题或者受害者有关。

（2）组合算法：使用多个单词或字符组合生成域名，这些单词或字符组合可能是攻击者自己定义的，也可能从外部数据中获取。这种 DGA 的域名通常是没有意义的，可能会包含随机字符或数字。

（3）神经网络算法：使用神经网络模型进行域名生成，该模型通常会使用大量的训练数据进行训练，以学习生成域名的规律。这种 DGA 的域名生成更加随机和复杂，因为其不仅依赖于种子和算法，还依赖于训练数据和模型的复杂度。

（4）混合算法：结合了多种算法进行域名生成，以增加随机性和安全性。例如，可以将时间戳种子和字典算法结合使用，生成具有一定意义的随机域名。

2.2 向量化方法

词嵌入（Word Embedding）是一种将文本中的单词映射到低维向量空间中的技术，可以将单词转化为向量，使得计算机可以更好地处理和理解自然语言。具体而言，词嵌入模型使用神经网络等机器学习技术，对文本中的单词进行处理，将单词映射到低维向量空间中的连续向量，使得相似的单词在向量空间中距离比较接近，不相似的单词则距离较远。词嵌入模型的训练需要大量的文本数据，一般使用语料库进行训练。语料库的选择和规模对于词嵌入的质量和效果具有重要影响。通常情况下，语料库的规模越大、质量越高，得到的词嵌入也越准确、具有更好的泛化性能。在使用词嵌入进行自然语言处理时，通常可以将词向量输入到神经网络中进行训练，也可以使用预训练好的词嵌入模型来提取文本特征，从而进行下游任务的处理。预训练好的词嵌入模型通常具有广泛的应用场景，并可以在各种自然语言处理任务中发挥作用。这些向量可以用于计算单词之间的相似度，以及用于文本分类、情感分析、机器翻译等任务中。常见的词嵌入算法包括 Word2Vec、GloVe 等，通过处理大量的文本数据，自动学习单词之间的关系，从而得到更准确、更有效的词嵌入表示。

2.2.1 独热编码

独热编码是一种简单而常用的词嵌入方法，将每个单词表示为一个高维稀疏向量，其中向量的每个维度对应于词汇表中的一个单词。在这个向量中，只有一个维度的值为 1，其余维度的值都为 0。这个值为 1 的维度被称为“one hot”维

度，因为这个向量在这个维度上只有一个“hot”元素，而其他维度都是“cold”的。例如，如果有一个包含 4 个单词的词汇表，如["apple", "banana", "orange", "pear"]，那么单词"apple"可以表示为[1, 0, 0, 0]，单词"banana"可以表示为[0, 1, 0, 0]，以此类推。这种方法非常简单，但没有捕捉到单词之间的语义关系，也没有考虑到单词在上下文中的含义。因此，在实际应用中，独热编码通常被更复杂的词嵌入方法所替代。

2.2.2 Word2Vec 向量化

Word2Vec 是一种常用的词嵌入方法，是由 Google 团队在 2013 年提出的。Word2Vec 通过使用神经网络来学习单词的嵌入表示，从而将单词映射到连续的向量空间中。Word2Vec 包含两种不同的模型，如图 2.2 所示：

CBOW(Continuous Bag of Words)：该模型通过上下文单词来预测中间的目标单词，将上下文单词的嵌入向量取平均后作为输入，通过一个全连接层来预测目标单词。

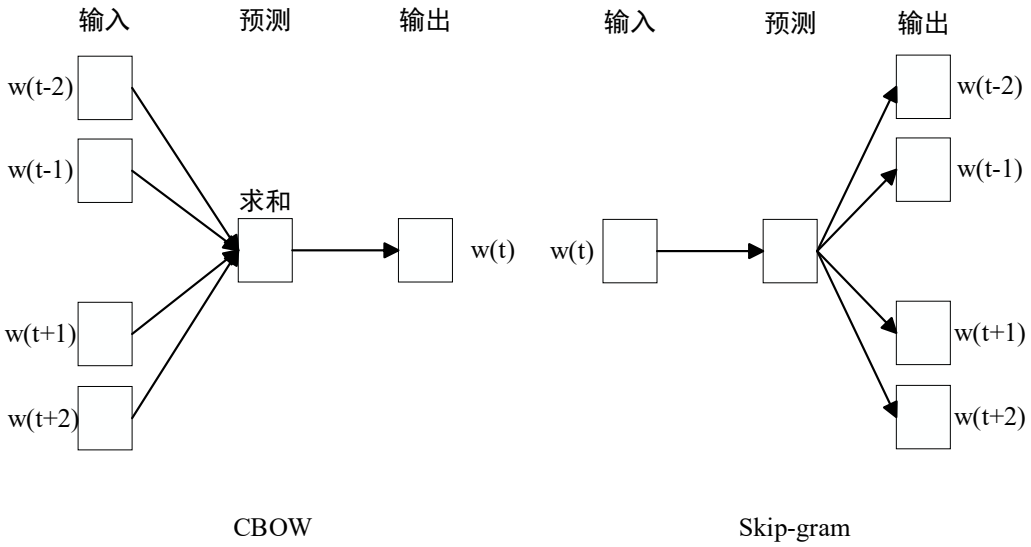

图 2.2 CBOW 和 Skip-gram 模型

Skip-gram：该模型通过中间的目标单词来预测上下文单词，将目标单词的嵌入向量作为输入，通过一个全连接层来预测上下文单词。

在训练过程中，Word2Vec 通过最大化目标单词和上下文单词之间的余弦相似度来优化模型，从而使得具有相似语义的单词在向量空间中距离更近。训练完成后，每个单词都可以表示为一个向量，这些向量可以用于自然语言处理任务，例如文本分类、情感分析、文本生成等。

Word2Vec 相对于传统的词袋模型和 One-Hot 编码，能够更好地表达单词之间的语义关系，并且具有较高的计算效率和可扩展性。目前，Word2Vec 已经成为自然语言处理领域中最为流行的词嵌入方法之一。

2.3 深度学习技术

2.3.1 卷积神经网络

卷积神经网络是一种常用的深度学习模型，常用于图像处理领域。卷积神经网络的核心是卷积层（Convolutional Layer），通过滑动一个可学习的卷积核（kernel）在输入数据上进行卷积运算，从而提取输入数据的局部特征。卷积层通常会结合非线性激活函数（如 ReLU）来引入非线性，增加网络的表达能力。在卷积层之后，通常会接一些池化层（Pooling Layer），用于缩减特征图的尺寸，减少模型参数和计算量，并提高模型的鲁棒性。在卷积层和池化层之后，通常会添加全连接层（Fully Connected Layer）和输出层，用于进行分类、回归等任务。卷积神经网络的训练通常采用反向传播算法，通过最小化损失函数来学习网络参数。常用的优化算法包括随机梯度下降及其变种（如 Adam、RMSprop 等）。卷积神经网络的优点在于可以自动学习特征，无需手工设计，且具有一定的平移不变性和局部不变性，适合处理图像等具有局部结构的数据。卷积神经网络也可以用于处理文本分类问题，特别是在短文本分类任务中，卷积神经网络已经证明了其高效性和有效性[51, 52]。

下面是使用卷积神经网络进行文本分类的一般步骤：

数据预处理：对原始文本数据进行清洗、分词、词干提取等预处理操作，以便后续处理。

向量化表示：将处理后的文本转换成向量表示，可以使用词袋模型、TF-IDF 等方法，也可以使用预训练的词向量（word embeddings）作为输入。在使用词向量作为输入时，可以选择不同的预训练模型（如 Word2Vec、GloVe 等）和不同的维度。

构建卷积神经网络模型：卷积神经网络模型通常由卷积层、池化层、全连接层等组成。在文本分类任务中，卷积层通常采用多个不同大小的卷积核来提取不同长度的特征，池化层用于降低特征维度，全连接层用于分类。

模型训练：将特征向量和标签输入到卷积神经网络模型中，通常使用随机梯度下降等优化算法学习网络参数。同时，还需要选择合适的损失函数（如交叉熵、Focal Loss 等）进行训练，以便在训练数据上获得最佳的分类性能。

模型评估：使用测试数据集对训练好的模型进行评估，评价指标包括精度、召回率、F1 值等。如果模型性能不佳，可以尝试调整模型超参数（如卷积核大小、池化层类型、学习率等）或增加训练数据量等方法进行调优。

总体来说，使用卷积神经网络进行文本分类需要进行数据预处理、特征提取、模型构建、模型训练和模型评估等步骤。在恶意域名分类任务中，还需要根据具体情况进行模型选择和调参等工作，以获得最佳的分类性能[53]。

（1）卷积层

如图 2.3 所示，卷积层是 CNN 的主要组成部分，通过使用卷积操作在输入数据上提取特征。卷积操作是指在输入数据上应用一个卷积核来提取特定的特征。卷积核是一个小的二维张量，其大小通常是正方形或矩形，其中包含了一些权重参数。在卷积操作中，卷积核会在输入数据上进行滑动，计算卷积核与当前滑动位置上的输入数据的乘积和，并把这些乘积和相加得到一个标量，最终组成了输出数据的一个像素。通过改变卷积核的权重，卷积操作可以提取不同的特征。

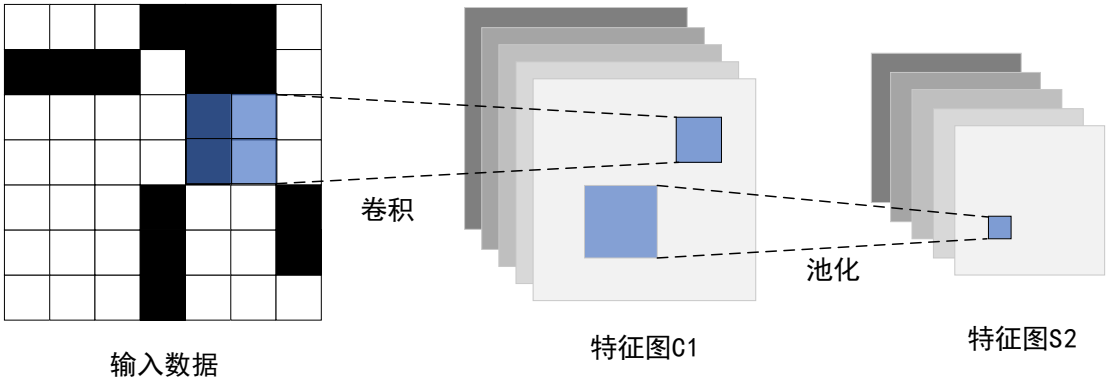

图 2.3 卷积操作示意图

卷积层中还包括一些超参数，如卷积核大小、步长（stride）、填充（padding）等。其中，卷积核大小指的是卷积核的高度和宽度，步长指的是卷积核每次滑动的距离，填充指的是在输入数据的边缘填充一些空白的像素，以使得输出数据的大小与输入数据相同或与其成比例关系。卷积操作后的图像大小计算方法如公式(2.1)所示。

$$N=\frac{W-F+2P}{S+1} \tag{2.1}$$

假设输入的图像长宽相同，其中，W 为输入图像的大小，F 为卷积核的尺寸大小，P 表示边界，S 则为步长的数值。

（2）激活函数

Sigmoid：Sigmoid 函数是是一种常用的激活函数，将输入的实数映射到 0 和 1 之间的范围内。其公式如(2.2)所示：

$$f(x)=\frac{1}{(1+e^{-x})} \tag{2.2}$$

其中 x 为输入的实数。Sigmoid 函数具有将任意实数映射到 0 和 1 之间的能力，且其输出值可以看作概率的估计值，通常用于二元分类问题中作为输出层的激活函数。

Sigmoid 函数的主要优点是可导的，并且可以对输出进行严格的限制，但也存在一些缺点。其中一个主要缺点是，当输入非常大或非常小时，Sigmoid 函数

的导数会变得非常小，这被称为梯度消失问题，如图 2.4 所示。此外，在深度神经网络中，Sigmoid 函数在反向传播过程中会导致梯度的爆炸和消失，从而使得模型的训练变得更加困难。因此，现在通常更倾向于使用其他的激活函数，如 ReLU 函数和其变种。

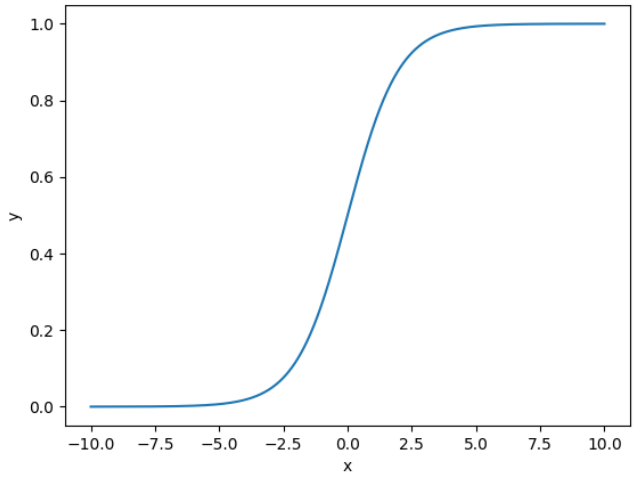

图 2.4 Sigmoid 函数

Softmax：Softmax 激活函数是一种用于将一组实数转换成概率分布的函数。在机器学习和深度学习中，Softmax 函数通常被用于将输出转换成一组概率，其中每个概率表示一个类别的概率，如图 2.5 所示。

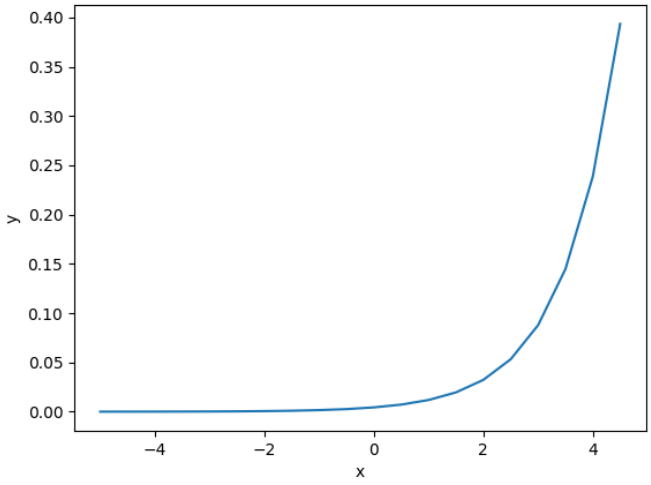

图 2.5 Softmax 函数

具体地说，给定一个包含 K 个实数的向量，Softmax 函数的输出为一个 K 维向量，其中每个元素的计算公式如(2.3)所示：

$$y_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2.3}$$

其中，e^x表示自然指数函数。Softmax 函数的主要特点是将输入转换为一个概率分布，其中所有概率都在 0 到 1 之间，并且所有概率的总和等于 1。在神经网络中，Softmax 函数通常被用于多分类任务，例如图像分类、自然语言处理中的词性标注、文本分类等任务中。

（3）池化层

池化层又被称为下采样层，是 CNN 中的重要组成部分，用于减小特征图的空间维度。池化层通常紧跟在卷积层之后，能够减少计算量，同时保留重要的特征信息，避免过拟合，有助于提高模型的泛化能力。通过对特征图中某一区域内的特征值进行汇聚，来获得整个区域的信息。常见的池化方式有最大池化（Max Pooling）和平均池化（Average Pooling）两种，如图 2.6 所示。通常情况下，池化层不会改变特征图的通道数，只会减小特征图的高和宽。

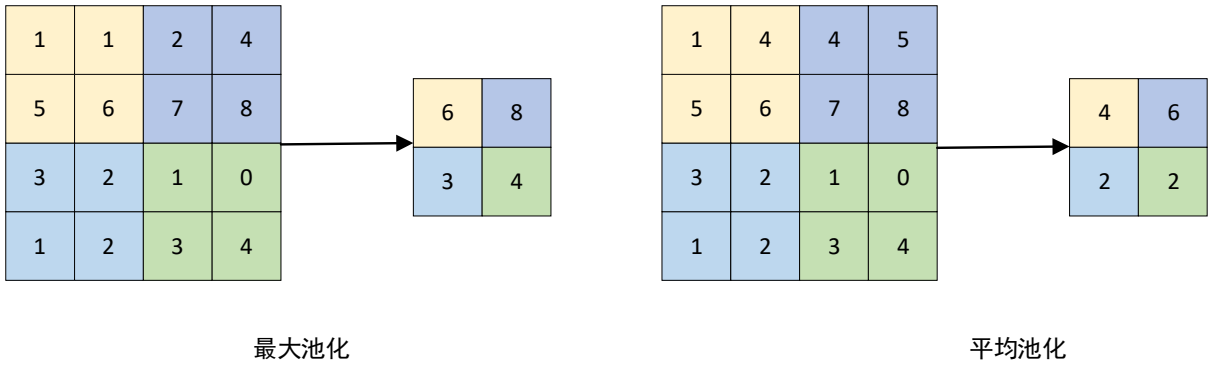

图 2.6 池化过程

最大池化层会在每个池化窗口中选择一个最大的值作为输出，因此可以保留特征图中最显著的特征，常用于提取边缘、角点等特征。而平均池化层则会在每个池化窗口中计算特征值的平均值作为输出，常用于提取纹理等整体特征。池化层可以有效地降低模型的计算复杂度，减小过拟合的风险，同时也可以帮助模型提取更具有代表性的特征。

（4）全连接层

全连接层是深度神经网络中的一种常用层。全连接层的输入是上一层的输出，每一个神经元都与上一层的所有神经元相连。在全连接层中，每个神经元都对上一层的所有神经元进行加权求和，并经过激活函数的处理，最终输出一个标量或向量。全连接层通常用于最后的分类或回归任务中，也可以用于将特征降维，提高模型的计算效率。

如图 2.7 所示，在卷积神经网络模型中，全连接层通常紧跟在卷积层和池化

层之后，是网络的最后一层。全连接层从展平层获取一维的输入数据，之后传入到包含非线性函数的全连接层中，最终隐藏层的输出发送到 Softmax 或 Sigmoid 函数，用于在类别上的概率分布。

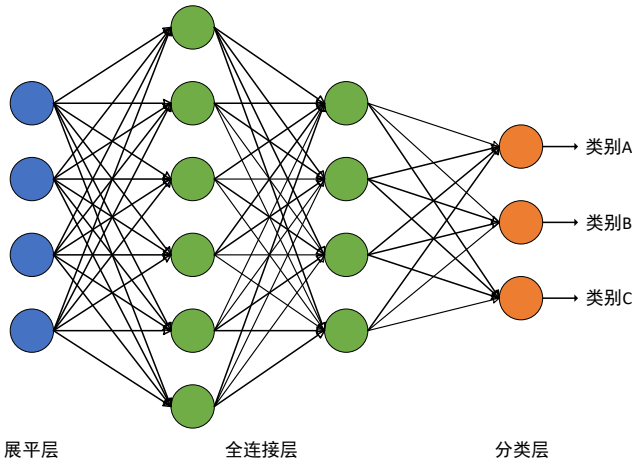

图 2.7 全连接层结构图

全连接层被用于将高维特征图映射到分类器中，以便输出最终的分类结果。全连接层的主要优点是可以建立多个神经元之间的复杂非线性关系，从而更好地适应数据的特征。但是需要大量的参数，容易造成过拟合，同时计算量也很大。因此，在域名分类中，常常会采用一些结构设计，如 dropout、正则化等技术来降低过拟合的风险。

2.3.2 深度可分离卷积

深度可分离卷积（Depthwise Separable Convolution）是一种针对卷积神经网络的改进技术。与传统的卷积操作相比，深度可分离卷积能够显著降低模型的计算量和参数量，并且在保持模型准确度的同时，可以提高模型的计算速度和推理速度。

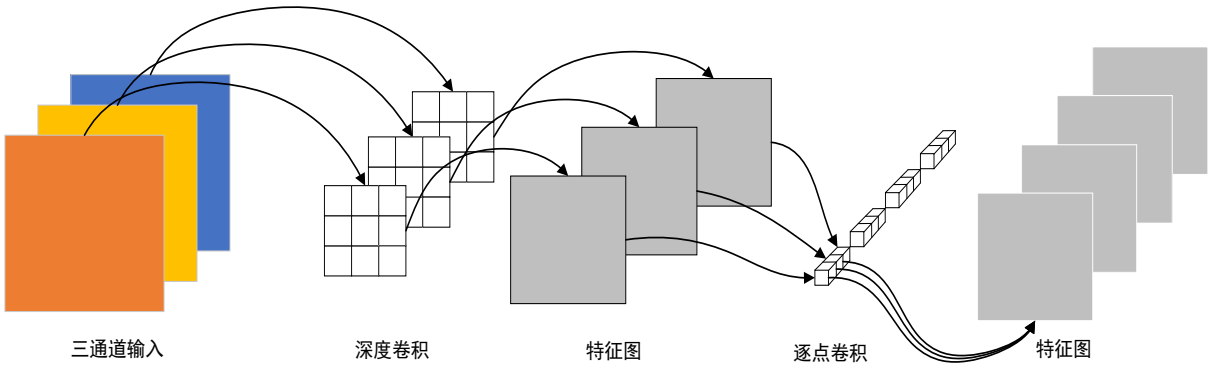

图 2.8 深度可分离卷积结构图

如图 2.8 所示，深度可分离卷积的原理是将卷积操作分解为深度卷积（Depthwise Convolution）和逐点卷积（Pointwise Convolution）。深度卷积操作

是对输入数据的每个通道分别进行卷积操作，得到多个通道的输出特征图；逐点卷积操作是对输出特征图进行卷积操作，以得到最终的输出结果。相比传统卷积操作，深度可分离卷积只需要对每个通道的卷积核进行处理，减少了模型的参数数量和计算量。

深度可分离卷积在计算机视觉和自然语言处理等领域都具有广泛的应用，例如图像分类、目标检测、文本分类等。在实际应用中，常用的深度可分离卷积模型包括 MobileNet、Xception 等。这些模型具有较小的模型大小和计算量，可以在资源受限的情况下快速进行训练和推理。随着深度学习技术的不断发展，深度可分离卷积模型在各个领域的应用也越来越广泛。

2.4 损失函数

在深度学习中，损失函数（Loss Fuction）是衡量模型预测结果与实际标签之间差异的一种函数。损失函数是机器学习中的重要组成部分，通常用于指导优化算法（如梯度下降）调整模型参数，以使得预测结果与实际标签之间的差异最小化。通常，损失函数的计算需要考虑模型的预测结果和实际标签之间的差异，以及模型对数据的拟合程度。常见的损失函数包括交叉熵损失函数（Cross Entropy Loss）、聚焦损失函数（Focal Loss）、标签平滑等。

以均方误差为例，假设模型的预测结果为 y_i，实际标签为 t_i，则均方误差如公式(2.4)所示：

$$L(y,t)=\frac{1}{2}\sum_{i=1}^{n}(y_i-t_i)^2 \tag{2.4}$$

其中，n 表示样本数量，y_i 和 t_i 分别表示第 i 个样本的预测结果和实际标签。通过计算损失函数，可以得到模型预测结果与实际标签之间的差异程度。在训练过程中，优化算法会根据损失函数的大小来调整模型参数，从而使得预测结果与实际标签之间的差异最小化。

总之，损失函数用于衡量模型预测结果与实际标签之间的差异，并指导优化算法调整模型参数。不同的损失函数适用于不同的问题和模型，选择合适的损失函数可以提高模型的准确度和鲁棒性。

2.4.1 交叉熵损失函数

交叉熵损失函数是一种常用于监督学习中分类问题的损失函数，可以衡量模型输出与真实标签之间的差异。

对于一个有 n 个类别的分类问题，假设模型输出的概率分布为 $P=(P_1,P_2,P_3...P_n)$，真实标签为独热编码的向量 $Y=(y_1,y_2,y_3...y_n)$，其中 $y_i=1$ 表示真实标签属于第 i 个类别，$y_i=0$ 表示不属于第 i 个类别。交叉熵损失函数如公式(2.5)

所示：

$$loss = -\sum_{i=1}^{n} y_i \log p_i \tag{2.5}$$

其中，log 为自然对数，p_i为模型输出的第 i 个类别的概率。

交叉熵损失函数的含义是，当模型输出的概率分布与真实标签的 one-hot 编码相同时，损失函数取最小值为 0；当两者分布不一致时，损失函数值大于 0，且差异越大，损失函数值越大。因此，交叉熵损失函数可以用来衡量模型分类预测的准确性，并且可以用于梯度下降等优化算法的反向传播过程中。

2.4.2 聚焦损失函数

Focal Loss 是一种用于解决类别不平衡问题的损失函数。在许多实际问题中，数据集中各类别的样本数量往往是不均衡的，这时使用传统的交叉熵损失函数可能会导致模型偏向数量较多的类别，而无法充分识别数量较少的类别。

Focal Loss 通过引入平衡因子和衰减因子的方式，改进了传统损失函数在类别不平衡情况下的性能。平衡因子可以降低数量较多的类别的权重，而衰减因子可以降低模型对容易分类的样本的关注度，提高对难分类样本的关注度，从而增强模型对数量较少的类别的识别能力。

Focal Loss 的表达式如公式(2.6)所示：

$$FL(p_t) = -\alpha_t (1-p_t)^{\gamma} \log(p_t) \tag{2.6}$$

其中，P_t表示模型对样本分类为正确类别的概率，α_t表示平衡因子，γ表示衰减因子。当γ=0 时，Focal Loss 退化为传统的交叉熵损失函数。Focal Loss 在处理类别不平衡问题时，具有较好的效果和实用性，已经在许多文本分类、目标检测等领域得到了广泛应用。

2.4.3 标签平滑

标签平滑（Label Smoothing）是一种正则化技术，常用于神经网络的分类任务中。在传统的分类任务中，标签通常使用独热编码，即将正确的类别设置为 1，其余为 0。然而，这种方式会使得模型计算交叉熵损失时，只有正确标签的维度参与了损失的计算，没有考虑其他标签位置的损失计算。通过这样方式的计算，会使得正确标签跟其他标签之间的关系被忽略，很多有用的信息无法捕获，使得模型泛化性较差，成为一个“非 0 即 1”的模型。这种计算 loss 的方式会导致模型在训练集上拟合的效果很好，在测试集上表现很差，容易过拟合。标签平滑则可以一定程度上缓解这个问题，通过将正确标签的概率降低一定量，将错误标签的概率增加一定量，将原始的独热标签转化为一个平滑的概率分布，减少模型对于训练数据中噪声和过拟合的敏感性。标签平滑公式如(2.7)所示，y_i为新的标签

向量，α为平滑因子，K为类别数。

$$y_i=\begin{cases}1-\alpha, & i=target\\ \alpha/(K-1), & i\neq target\end{cases} \tag{2.7}$$

若有一批数据集，数据集的类别总数为 6，随机抽取一个样本，该数据进行独热化编码后的标签为$[0,1,0,0,0,0]$。假设已经得到该数据进行 Softmax 的概率矩阵P，即:$P=[P_1,P_2,P_3,P_4,P_5,P_6]=[0.3,0.6,0.01,0.01,0.05,0.03]$。使用交叉熵损失函数可以求得当前数据的 loss 约为 0.22。

由于 One-Hot 编码只是对真实情况的简化，造成对标签类别不合理的表示。对于 One-Hot 带来的容易过拟合的问题，使用标签平滑的方式进行解决。现设一个平滑因子α=0.25，可以对该数据的标签进行如下改变，如公式(2.8)、(2.9)、(2.10)所示：

$$y_1=(1-\alpha)\times[0,1,0,0,0,0]=[0,0.75,0,0,0,0] \tag{2.8}$$

$$y_2=\alpha/(K-1)\times[1,0,1,1,1,1]=[0.05,0,0.05,0.05,0.05,0.05] \tag{2.9}$$

$$y_i=y_1+y_2=[0.05,0.75,0.05,0.05,0.05,0.05] \tag{2.10}$$

y_i就是经过平滑方法后得到的标签，平滑后该数据的交叉熵损失约为 0.53。平滑后的数据通过交叉熵损失函数不仅考虑了训练集中正确标签位置的损失，也将错误标签位置的损失考虑在内，从而使得最终的损失变大，驱动模型的学习能力提高。因此，模型不仅能增加正确分类的概率，还能减少错误分类的概率。标签平滑就是把原来 One-Hot 表示在每个维度都添加一个随机噪音，一定程度上可以缓解模型过于武断的问题，增加了信息量，提供了训练数据中类别之间的关系。标签平滑简单有效，把预测值过度集中在概率较大类别上，其他概率较小类别上也会有偏向。其优点为：可以缓解模型过于自信的问题，也有一定的抗噪能力；弥补了简单分类中信息熵较少的问题，增加了信息量；产生更好的校准网络，从而更好地泛化，能对未知的数据得到更准确的预测，在图像分类和文本分类中应用十分广泛。

2.5 性能评价指标

本文提出的恶意域名检测模型会将输入的待测域名进行合法或恶意的分类。评估模型的评价指标基于实验结果的混淆矩阵计算，显示模型在测试数据集上的预测结果和实际标签之间的差异。

真正例（True Positive，TP）表示实际标签为正例，模型预测也为正例的样本数量。假正例（False Positive，FP）表示实际标签为负例，模型预测为正例的样本数量。假反例（False Negative，FN）表示实际标签为正例，模型预测为负例的样本数量。真反例（True Negative，TN）表示实际标签为负例，模型预测也为负例的样本数量。如表 2.1 所示。

表 2.1 混淆矩阵

真实情况	预测情况	
	正例	反例
正例	TP	FN
反例	FP	TN

分类模型的性能评价指标包括以下几个方面：

准确率（Accuracy）：指分类模型预测结果正确的比例，如公式(2.11)所示。准确率越高，模型分类能力越好。

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.11}$$

精确率（Precision）：指模型预测为正例的样本中，真正为正例的比例，如公式(2.12)所示。精确率越高，模型预测正例的能力越强。

$$Precision = \frac{TP}{TP + FP} \tag{2.12}$$

召回率（Recall）：指模型正确预测为正例的样本占实际为正例的比例，如公式(2.13)所示。召回率越高，模型正确识别正例的能力越强。

$$Recall = TPR = \frac{TP}{TP + FN} \tag{2.13}$$

F1 分数（F1 Score）：是精确率和召回率的调和平均数，如公式(2.14)所示。F1 分数综合考虑了精确率和召回率的影响，适用于不平衡的数据集。

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.14}$$

FPR 是指在实际为负例的样本中，被错误预测为正例的样本占实际负例样本总数的比例，如公式(2.15)所示。FPR 的值越低，说明分类器在预测负例时表现越好，也就是说分类器更准确地排除了负例样本，避免将其误判为正例。在某些应用中，如异常检测和医学诊断中，FPR 往往是一个非常关键的性能指标，因为在这些应用中，误判一个正常样本为异常样本可能会造成更大的影响。FPR 通常与真正类率之间存在一种权衡关系。降低 FPR 的同时，往往会降低 TPR，而提高 TPR 的同时也往往会提高 FPR。因此，在评估分类器性能时，需要综合考虑 FPR 和 TPR 的权衡关系，选择适当的阈值和模型参数，以获得最佳的分类器性能。

$$FPR = \frac{FP}{FP + TN} \tag{2.15}$$

ROC（Receiver Operating Characteristic）：是真正类率与假正类率之间的关系图，通常以不同阈值下的 TPR 和 FPR 为坐标绘制。ROC 曲线的面积 AUC（Area Under the ROC Curve）可以衡量模型分类能力的优劣，如公式(2.16)所示，AUC

值越高，模型分类能力越好。

$$\text{AUC}=\int(TPR)d(FPR) \tag{2.16}$$

TPR@FPR 是指在 FPR 等于给定值时，TPR 的值。TPR@FPR 通常用来评估二分类模型在不同阈值下的性能表现，可以在一个给定的 FPR 下选择最佳的分类模型。

2.6 本章小结

本章主要介绍了使用深度学习进行恶意域名检测的相关基础理论和技术。第一节介绍了域名的相关知识。第二节介绍了对域名进行向量化表示的各种向量化方法。第三节介绍了对域名的向量化表示进行提取特征的卷积神经网络，以及对卷积神经网络改进的技术深度可分离卷积。第四节介绍了本文所使用的各种损失函数。第五节介绍了本文使用的性能评价指标。

第3章 基于字符特征和词特征融合的恶意域名检测

3.1 引言

在恶意域名检测任务中，现有模型大多基于单一字符、多维字符、单一词结构、多种词结构等特征，忽略了字符和词之间的关联语义。对于由随机字符组成的恶意域名检测效果较好，但是对由随机单词组成的 DGA 域名检测精度不高。针对上述问题，提出一种基于字符和词特征融合的恶意域名检测模型 CWNet。首先利用并行卷积神经网络分别提取域名中字符和词的特征；其次将两类特征进行拼接，构造成融合特征；最后输出层使用 Softmax 激活函数得到待测域名最大概率的分类预测结果。损失函数使用 Focal Loss，更加关注难分类的样本，提高模型分类的性能。实验结果表明，该算法可以提升对恶意域名的检测能力，对更具挑战性的恶意域名家族的检测准确率提升效果更为明显。

3.2 模型结构

3.2.1 模型构造

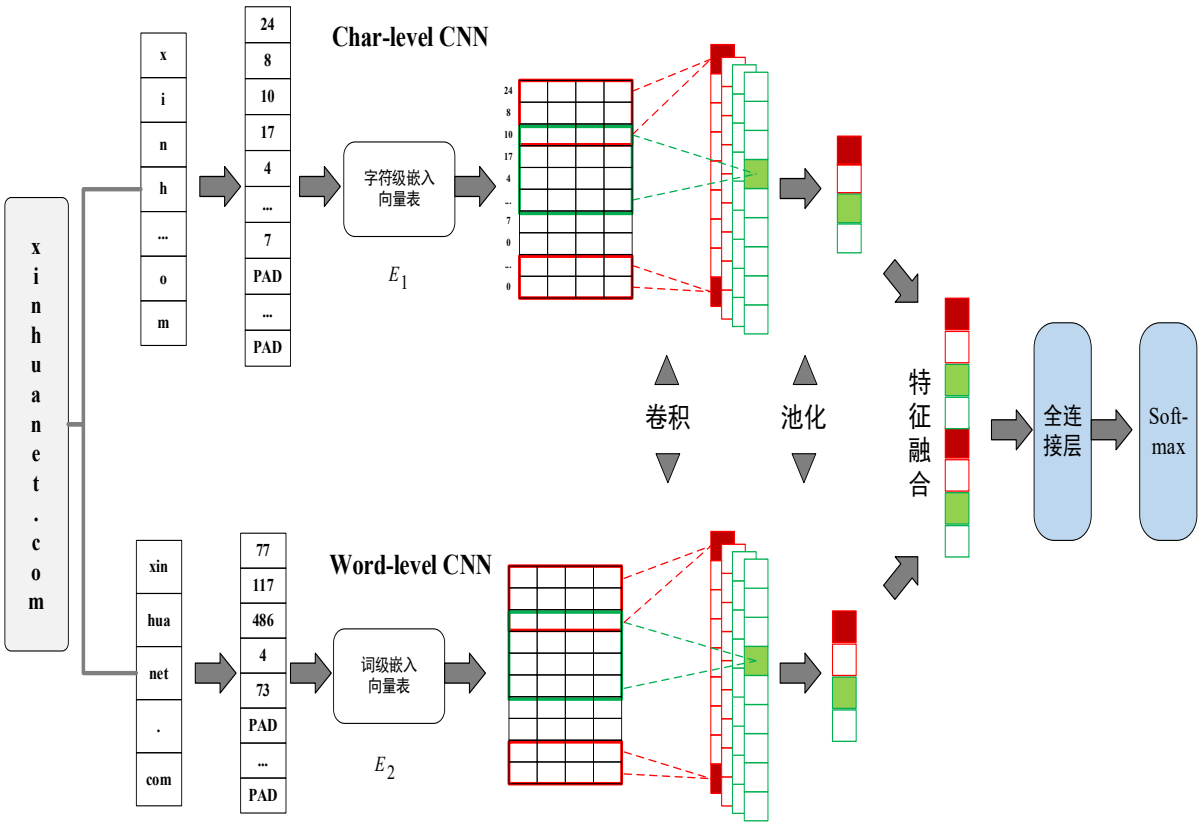

图 3.1 CWNet 网络结构图

如图 3.1 所示为本文提出的恶意域名检测模型的整体框架图。主要包括字符级特征提取、词级特征提取、特征融合等模块。首先，将域名字符串作为 CWNet 的输入，利用并行卷积神经网络分别提取域名的字符级特征和词级特征；然后，将两种特征进行拼接融合；最后，利用 Softmax 实现待测域名的分类[54]。

3.2.2 字符级特征提取

1.字符嵌入

由于域名不区分大小写，在数据预处理阶段将所有域名的大写字母转改为小写。此处，统计域名数据集中唯一出现的字母、数字和特殊字符，包括 26 个英文字母，10 个数字字符，连词号“-”，分隔符“.”，和设置的“PAD”、“UNK”，共获得 N=40 个不同字符。使用 Word2Vec 向量化技术对数据集中的字符进行向量化；将每个字符转化为 D 维的向量，D 设置为 30，存储在向量表 $E_1 \in \mathbb{R}^{N\times D}$ 中[55]。

如图 3.2 所示，通过统计数据集中最长域名的字符个数为 67，所以将字符级向量表示的长度 L_1 设置为 67，长度小于 67 的域名字符串使用零向量填充。利用该向量化方法将数据集中每条域名的单个字符 D_i 转换为 $d_i \in \mathbb{R}^{L_1\times D}$ 的向量表示，串联所有域名字符，获得整条域名的向量化表示[56]。

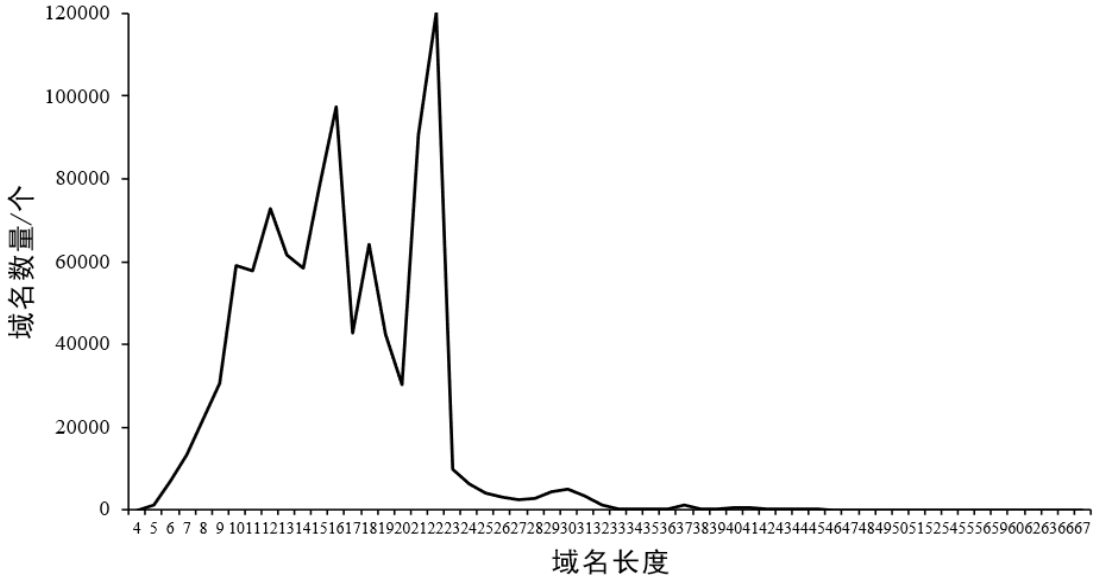

图 3.2 数据集域名长度统计图

2.字符特征提取

通过统计发现本文使用的域名数据集中最小域名字符串的长度为 4，最大长度为 67。设定卷积核 $W \in \mathbb{R}^{h\times D}$ 的大小为 2，3，4，5，其中 h 为卷积核的大小，每类卷积核的个数为 128[57]。

利用一维卷积对域名向量表示进行卷积操作，提取域名字符与字符之间的局部相关性，得到特征图后进行最大化池化并拼接各个池化值，得到域名的特征表示，如图 3.3 所示。

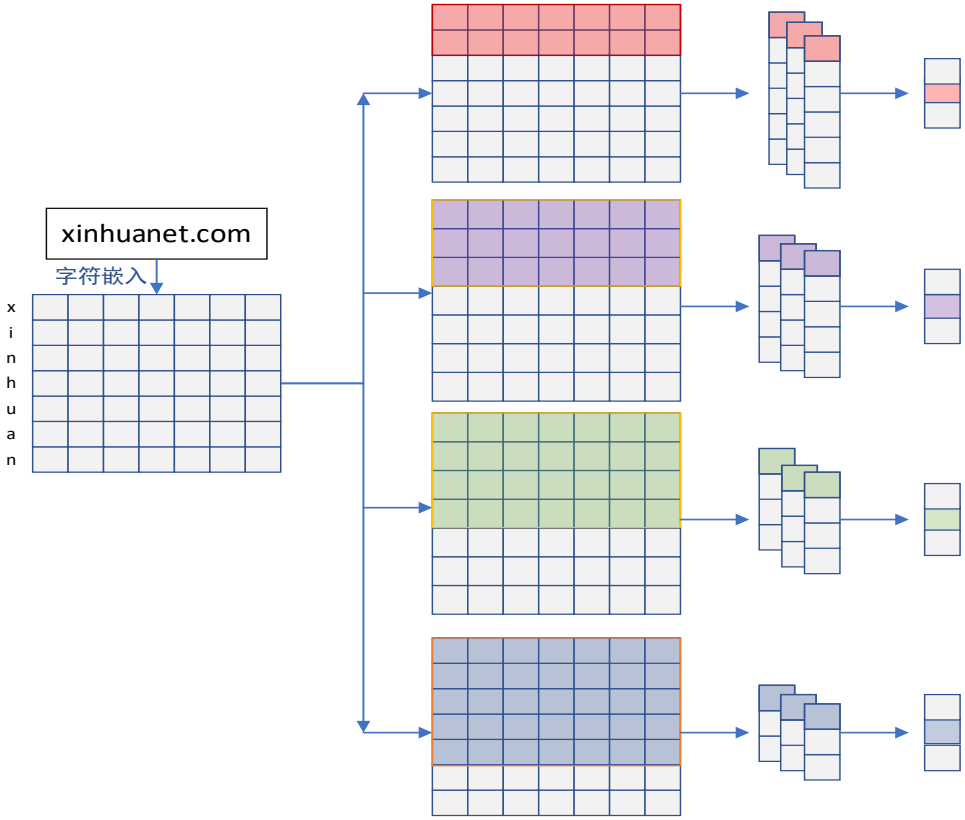

图 3.3 域名字符特征提取

3.2.3 词级特征提取

如表 3.1 所示，不同的 DGA 域名构造规则不同，根据 DGA 家族的域名生成规则可将其分为 2 类，一类是由随机字符组成的 DGA 域名，另一类是由随机单词组成的 DGA 域名。由随机字符组成的 DGA 域名由于具有高度的随机性特征，与合法域名存在较大的差异，其检测难度较为低。由随机单词组合而成的恶意域名，其组成方式是根据在预先构建的词典中随机挑选几个单词组合而成，在组成方式上与部分合法域名存在相似性，其检测难度较大。因此，单独的字符级特征不足以区分所有类型的域名，本文通过融合字符和词特征进行合法域名与恶意域名的分类。

1.词嵌入

（1）简单词嵌入

如表 3.1 所示，给出了部分由单词组成的恶意域名家族的构造规则，分别从词表或词典中随机挑选 2~4 个单词，并与对应的顶级域名组合形成 DGA 域名。为了对域名字符串进行分词处理，将字典、词表、顶级域名和字符级的 40 个字符进行组合，形成仅具有单一字符或单词的词表，并作为 wordninja 包的语料库。使用 wordninja 即可对域名字符串按照最长字符序列匹配方式进行分词，得到多个字符或单词的子串。词嵌入同样使用 Word2Vec 对词表中的字符和单词进行向量化，并存储在向量表 E_2 中。

表 3.1 恶意域名家族的构造规则

	DGA 家族	顶级域名	构造规则	示例
随机字符组合的 DGA 家族	blackhole	ru	固定长度 16，a-z	mkjdkbwuxcnuxtqd.ru ppxhzopqbiykuucv.ru tmlrjxvvrvkyxofn.ru
	ccleaner	com	长度 11-13，混合 a-f 和 0-9	ab1145b758c30.com ab890e964c34.com ab3d685a0c37.com
	chinad	com, org, net, biz, info, ru, cn	固定长度 16，混合 a-z 和 0-9	qowhi81jvoid4j0m.biz 29cqdf6obnq462yv.com 5qip6brukxyf9lhk.ru
	dmsniff	com, org, net, ru, in	固定长度 8，a-z	kojnwyop.org albdfhln.com enmspvru.net
随机单词组合的 DGA 家族	bigviktor	artdfg,click,club,com, fans 等，共 20 种	组合来自 4 个预定义词典的 3~4 个单词	remote-conclusion.fans keep-thinprofit.net stop-remotesky.org
	matsnu	com	组合来自 2 个预定义词典的 2~3 个单词	world-bite-care.com activitypossess.com mattermiss-type.com
	ngioweb	net, info, com, biz, org, name	组合来自 3 个词表的单词	interegoging.org ultrafarihood.com prodoboly.name
	suppobox	net,ru	组合来自 3 个词表的 2 个单词	tablethirteen.net childrencatch.net thinkgoodbye.ru

由于存在分词后仅有字符的域名，所以词级向量表示的长度 L_2 和维度 D 同样分别设置为 67 和 30，长度小于 67 的域名字符串使用零向量填充。利用该向量化方法将数据集中每条域名的单个单词 D_i 转换为 $d_i \in \mathbb{R}^{L_2 \times D}$ 的向量表示，串联所有域名字符，获得整条域名向量化表示。

以“support.showremote-conclusion.fans”恶意域名为例进行说明，如图 3.4 所示。首先，使用 wordninja 对域名字符串进行分词，得到多个单词或字符；然后，根据单词或字符的索引，从词级向量表 E_2 中得到对应的向量表示；最后，将单词和字符的向量表示拼接，得到域名最终的向量化表示 $x \in \mathbb{R}^{L_2 \times D}$。

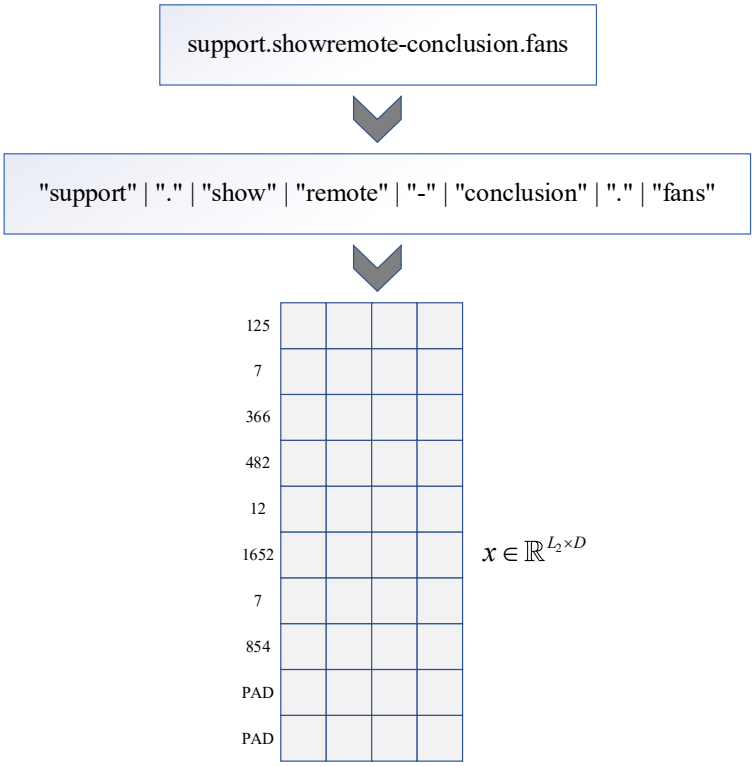

图 3.4 简单词嵌入

(2)字符级词嵌入

在提取词级特征表示后，将词作为分割单元，细粒度地提取单词中的字符级特征表示，并将词级嵌入和字符级嵌入进行融合。

如图 3.5 所示，通过统计字典、词表、顶级域名中单词的长度最长为 18，长度为 2~10 的占总数的 94.2%，为了减少模型参数量 L_3 设置为 10，L_3 为每个词被填充为的字符序列的大小，大于 10 的进行截断处理。

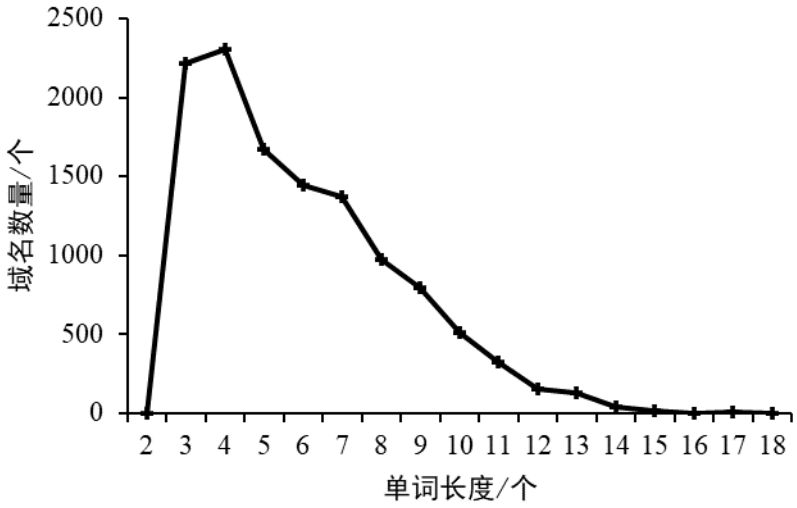

图 3.5 恶意域名词典单词长度统计

首先，获取域名的词向量表示 $x_1 \in \mathbb{R}^{L_2 \times D}$；然后，获取域名中每个单词的字符级向量表示 $L_3 \times D$，此处将每个词被填充为 L_3=10 的字符序列，将所有单词的字符级向量表示拼接得到 $x_2 \in \mathbb{R}^{L_2 \times L_3 \times D}$。最后，对所有词的字符向量求和，得到域名的

字符级嵌入 $x_2 \in \mathbb{R}^{L_2 \times D}$，并将 x_1 和 x_2 两个嵌入矩阵求和，得到最终的域名字符级词嵌入，如图 3.6 所示。

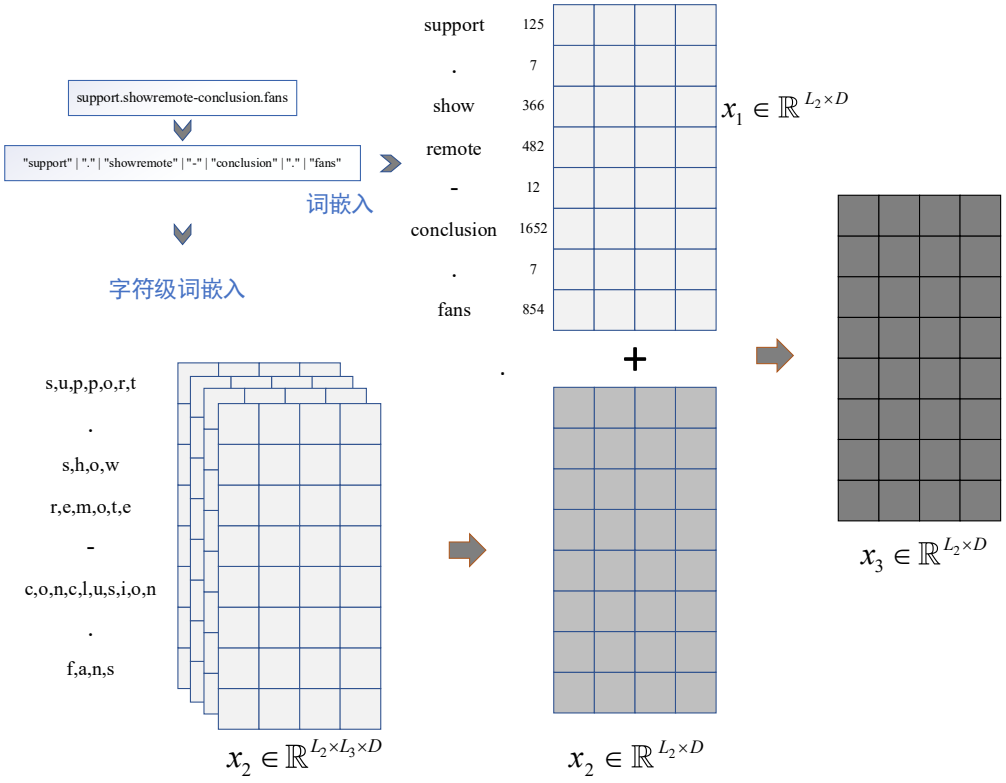

图 3.6 字符级词嵌入

2.词特征提取

由表 3.1 可知，随机单词组成的 DGA 域名通常包含 3 至 5 个单词。设定卷积核 $W \in \mathbb{R}^{h \times D}$ 的大小为 2、3，其中 h 为卷积核的大小，每类卷积核的个数为 128。

利用一维卷积对域名向量表示进行卷积操作，卷积核仅向下移动，提取域名词与词之间的局部相关性，得到提取后的特征图，然后对每个特征图进行最大化池化并拼接各个池化值，最终得到域名的特征表示，如图 3.7 所示。

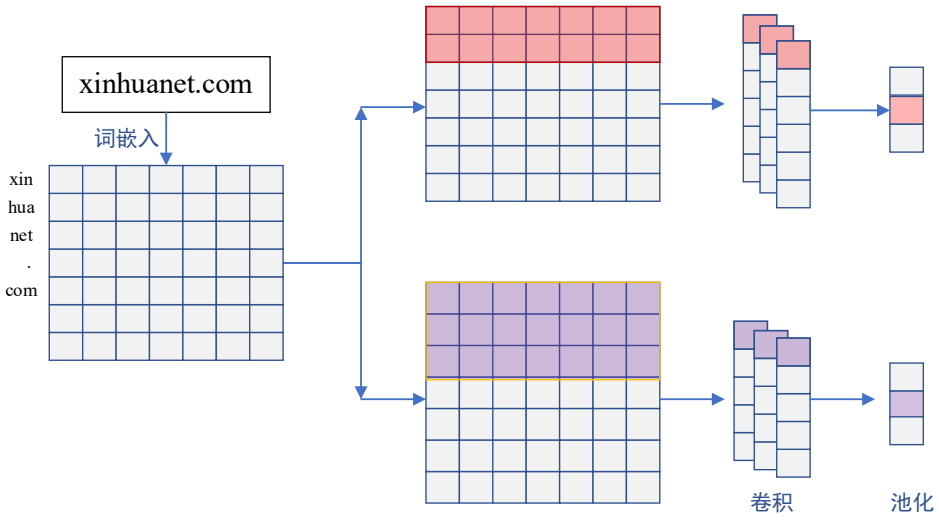

图 3.7 域名词级特征提取

3.2.4 特征融合

首先，字符级和词级部分在卷积操作之后，进行最大池化操作，分别得到域名字符和词的特征图。然后，将字符级和词级得到的特征图进行拼接分别接入 256 个结点的全连接层。最后，使用 concat 将字符和词的特征进行融合，作为之后全连接层的输入，如图 3.8 所示[58]。

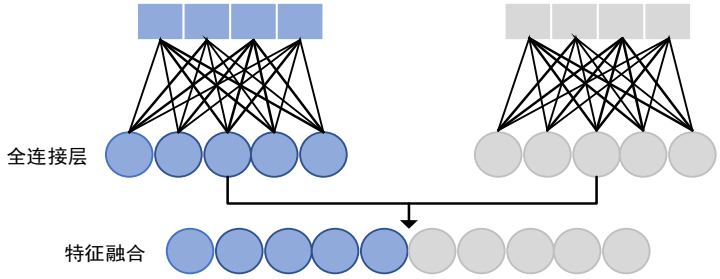

图 3.8 特征融合

3.2.5 全连接层

全连接层中每个结点都与前一层的所有结点进行连接，将融合特征向量接入 3 个全连接层，全连接层的结点数量分别为 256、128、64。最后输出层使用 Softmax 归一化指数函数进行处理，得到合法和恶意的概率预测值，其定义如公式(3.1)所示。

$$y_i = \frac{e^{Z_i}}{e^{Z_1} + e^{Z_2}}, i = 1,2 \tag{3.1}$$

其中，Z_i 是输入向量。输出向量 y_i 介于 0 和 1 之间，所有输出向量的和为 1。

3.2.6 聚焦损失函数

DGA 家族可分为由随机单词组成和随机字符组成的恶意域名两类。前者的形成规则与合法域名类似，是由可发音的词组成的域名，是较难检测的恶意域名。DGA 家族种类繁多，不同类别的 DGA 域名检测难度不同，恶意域名检测存在检测不均衡的问题。

为了减少简单样本对损失函数的影响，更加关注难分类的样本，所以引入 Focal loss 作为损失函数，提高模型分类的性能，如公式(3.2)所示。

$$FL = \begin{cases} -\alpha(1-y')^{\beta} \log y', y = 1 \\ -(1-\alpha) y'^{\beta} \log(1-y'), y = 0 \end{cases} \tag{3.2}$$

y 是样本的真实值，y' 为网络模型输出的预测结果值，α 用于调整正负样本不平衡的比例，β 用于使模型更加关注难以检测的样本。如图 3.9 所示，不同 β 条件下 FocalLoss 的 Loss 变化。β 越大，简单样本产生的损失权重越小，困难样本产生的损失权重越大。由于数据集正负样本比例为 1:1，根据实验结果，α 和 β 值分别设置为 0.5 和 2 可以得到最佳的效果[59]。

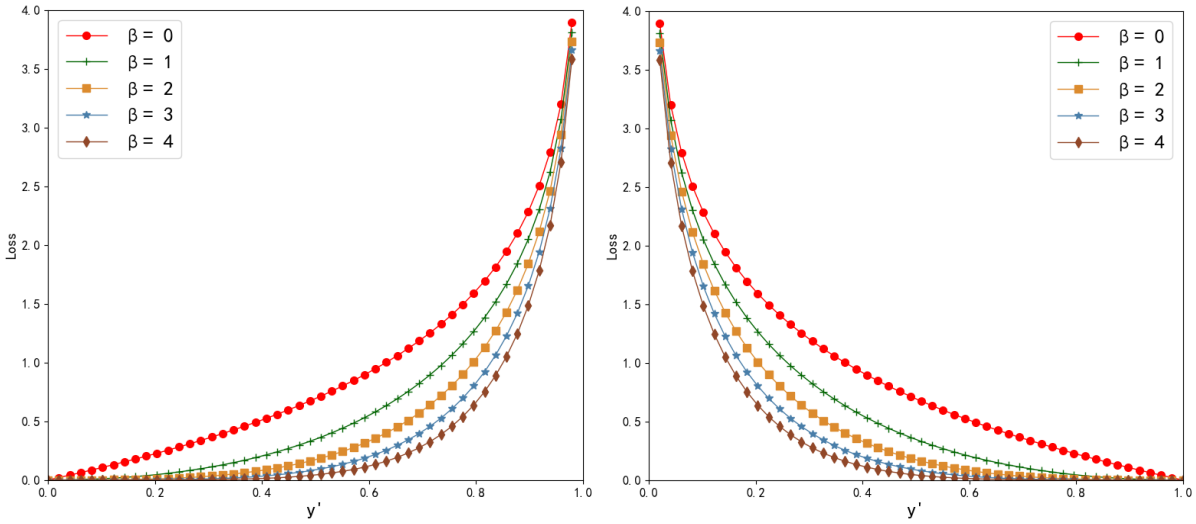

图 3.9 Focal Loss 损失函数

3.3 实验结果及分析

3.3.1 数据集

表 3.2 合法和恶意域名数据集描述

域名类型	来源	数目
恶意域名	360Netlab：abcbot，bigviktor，dircrypt，mydoom，qadars，ramnit 等 56 个 DGA 家族	498442
合法域名	Alexa 前 500000 条	500000

如表 3.2 所示，收集与整理 360 安全实验室开源的多家族恶意域名数据集，包括 suppobox，rovnix，tinba 等 56 个家族，共 498442 条恶意域名，设定该类域名的标签为 1；选取 Alexa 网站排名前 500000 条域名作为合法域名数据集，标签设置为 0。

并将所有数据按照通 3:1:1 的比例划分训练集、测试集和验证集，如表 3.3 所示。

表 3.3 合法和恶意数据集划分

	合法域名	恶意域名	总量
训练集	300000	299065	599065
验证集	99889	99800	199689
测试集	100111	99577	199688
总量	500000	498442	998442

3.3.2 实验环境及评价指标

本章节实验的实验环境如表 3.4 所示。

表 3.4 第 3 章实验环境表

实验环境	参数
CPU 处理器	AMD Ryzen 7 5800H 3.30GHz
GPU	NVIDIA GeForce RTX 3060 6GB
内存	16GB
操作系统	Windows11
加速库	CUDA11.0
深度学习框架	Pytorch1.7.1
编程工具	Pycharm

对比实验采用的评价标准包括准确率（Accuracy），召回率（Recall），精确率（Precision），F1 值（F1-score）和 FPR（False Positive Rate）五项指标。由于阻止合法域名是不可取的，因此不仅要关注模型的检测准确率，低的误报率也是非常重要的。

消融实验采用的评价标准为 TPR@FPR，AUC。TPR@FPR 可以明确地展示在相同 FPR 条件下本文模型不同网络的 TPR 取值情况。AUC 表示 ROC 曲线下的区域，可以准确反映 TPR 和 FPR 的关系，是检测模型准确性的综合代表。

3.3.3 对比实验

如表 3.5 所示，本文模型与以下五种恶意域名检测模型在相同的数据集进行了对比实验。

表 3.5 五种深度学习对比模型

模型结构	参考文献
LSTM	[34]
CNN + BiGRU	[39]
DCNN	[38]
Invincea	[36]
Improved Invincea	[37]

在准确率，召回率，精确率，F1 值，误报率五种评价指标下进行分析，实验结果如表 3.6 所示。并且分别在 10 种 DGA 家族的数据集下测试准确率，实验结果如图 3.10 所示。

将 CWNet（Char），CWNet（Char-Word）以及将两者结合起来的 CWNet（Char+Char-Word）模型与上述模型进行了比较。相比之下，CWNet 均有明显的提升。与性能最好的对比模型相比，准确率提升了 1.60%，召回率提升了 2.28%，

精确率提升了 0.98%，F1 值提升了 1.63%，误报率也下降了 0.94%。

表 3.6 对比实验结果

模型	准确率	召回率	精确率	F1 值	误报率
LSTM	0.9403	0.9298	0.9494	0.9395	0.0493
CNN + BiGRU	0.9412	0.9310	0.9501	0.9405	0.0487
DCNN	0.9557	0.9426	0.9679	0.9551	0.0311
Invincea	0.9665	0.9540	0.9783	0.9660	0.0210
Improved Invincea	**0.9703**	**0.9593**	**0.9807**	**0.9699**	**0.0188**
CWNet(Char)	0.9813	0.9782	0.9842	0.9811	0.0156
CWNet(Char-Word)	0.9702	0.9591	0.9807	0.9697	0.0187
CWNet(Char+Char-Word)	**0.9863**	**0.9821**	**0.9905**	**0.9862**	**0.0094**

如图 3.10 所示，构建了 10 个黑名单数据集，分别是由随机字符组成的 cryptolocker，dyre，ramnit，shiotob，necro，qadars，virut 这 7 个 DGA 家族和 ngioweb，suppobox，matsnu 这 3 个较难检测的由随机单词组成的 DGA 家族。本文的 CWNet（Char+Char-Word）模型分别和上述五个对比模型对这 10 个黑名单进行准确率测试。

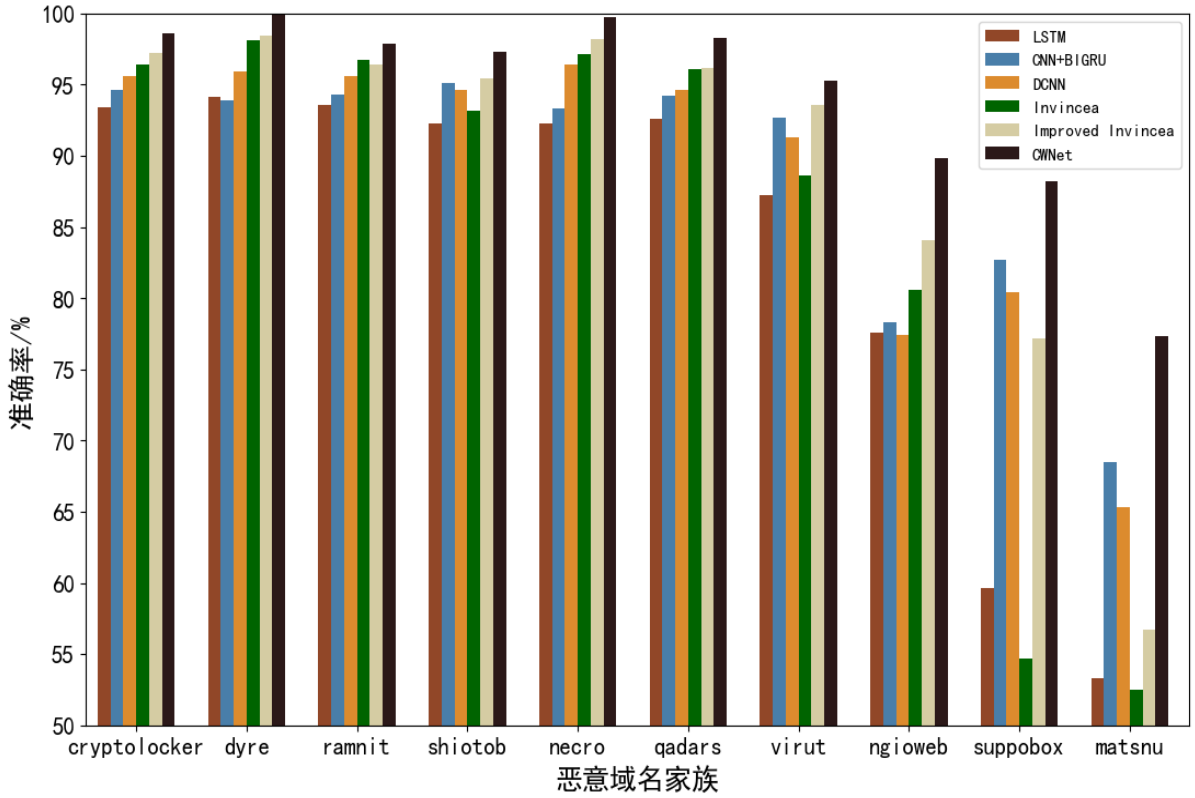

图 3.10 黑名单数据集准确率测试图

本文模型在 10 个黑名单中均得到最高的准确率，在 dyre 家族的检测准确率达到 100%，且在较难检测的 ngioweb，suppobox，matsnu 家族的准确率明显高于其他模型，进一步验证了本文模型使用字符和词的融合特征检测恶意域名的有效

性。

由实验结果分析可知，现有模型基于 LSTM、CNN 的方法仅是利用了域名字符串中的局部特征或序列时序特征，对于随机字符组成的 DGA 域名有较好的检测效果，但是对于随机单词组成的 DGA 域名很难得到区别于合法域名的局部特征。CWNet 能够捕获域名字符串中的不同长度的字符级和词级信息，在网络模型中加入字符和词的特征融合部分，将域名中的字符特征和词特征进行融合，可有效提高域名文本信息的利用度。

3.3.4 消融实验

为了验证本文模型不同网络对域名分类的影响，在测试集上进行消融实验，使用 TPR@FPR 以及 AUC 作为评价标准。CWNet 不同网络的测试性能如表 3.7 所示。

表 3.7 消融实验结果

	TPR@FPR Level				AUC
	0.0001	0.001	0.01	0.1	
(CWNet)Character-level:					
Char CNN	0.6367	0.9294	0.9778	0.9976	0.9986
(CWNet)Word-level:					
Word CNN	0.5661	0.8010	0.9080	0.9699	0.9765
Char-Word CNN	0.6347	0.9652	0.9427	0.9890	0.9957
(CWNet)Whole:					
Char+Word CNN	**0.7961**	0.9314	0.9796	0.9978	0.9987
Char+Char-Word CNN	0.7196	**0.9392**	**0.9830**	**0.9981**	**0.9988**

在词级部分，使用字符级词嵌入可以提高 AUC，验证了细粒度地提取单词中的字符级特征与词级特征融合可以更有效地表示单词。CWNet（Whole）利用字符级和词级融合特征显著优于 CWNet（Character-level）和 CWNet（Word-level），而且还比较了模型在不同误报率水平下的召回率，在所有指标都展现了更好的性能。

CWNet（Char+Char-Word）能够捕获域名字符串中的不同长度的字符级和词级信息，而且使用字符级词嵌入得到的词特征与其字符特征进行融合，可以获得更有效的域名特征。与单一的字符级或词级网络相比，域名文本信息的利用度更高。

本文所提出的 CWNet（Char+Char-Word）是 CWNet（Char CNN），CWNet（Char-Word CNN）并行组成的模型，其 AUC 可视化如图 3.11 所示。

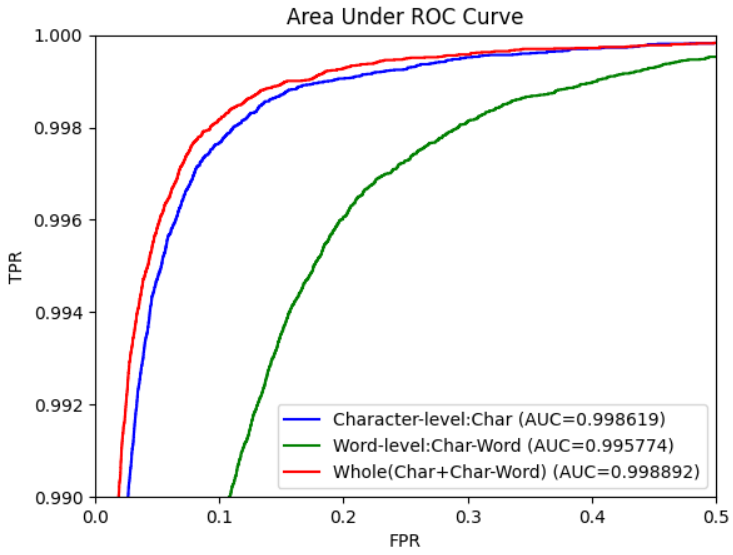

图 3.11 字符级词级和特征融合模型消融实验 AUC

3.4 本章小结

本章提出了一种基于字符级和词级特征融合的恶意域名检测模型 CWNet，通过提取字符特征和词级特征并进行特征融合，与现有模型相比，对域名字符串所提供的信息利用度更高。通过在开源数据集上进行测试，结果验证了本文模型的有效性，表明字符和词融合特征的充分利用可以有效提高对 DGA 域名的检测性能，且对随机单词组成的 DGA 域名的检测性能也有明显的提升。虽然该网络可以提供良好的检测精度，但是并行卷积神经网络模型参数较多，大量可训练的参数导致较高的内存消耗，应用难度较大。为了解决这一问题，后续将设计更加轻量化的恶意域名检测模型。

第4章 基于全卷积的快速恶意域名检测

4.1 引言

由于移动设备和边缘设备快速发展，将人工智能应用于受限平台已经成为一种趋势。目前受限平台的处理性能还不能够训练庞杂的深度学习模型，但现在已经可以通过在平台上配置训练好的模型来进行推理。这种方法的优势有很多，例如无需实时网络依赖且响应速度够快。尽管如此，在受限平台上使用卷积神经网络模型仍然受到其存储空间大小的限制。在卷积神经网络模型上的研究大多都着重于增加网络深度以提高性能。在恶意域名检测领域，已经提出了一些性能较好的深度模型，但这些模型是具有大量参数的体系结构，占用内存空间较大，无法平稳运行在受限平台。

上一章节中介绍了一种使用并行卷积神经网络将域名的字符和词特征进行融合的恶意域名分类模型。该模型使用并行卷积神经网络提取域名特征的效果非常优良且使用全连接层可以对待测域名有更高的分类精度，但由于使用了多个不同尺寸的卷积核对域名向量化表示进行特征提取，使得模型较为复杂，之后的全连接层更是让模型整体的计算量和参数数量进一步提升。由于此模型计算量和参数量过多，很难在现实场景中得到应用。

上述产生的问题其根本原因是模型训练的参数数量过多。本章节在上一章节所提出的 CWNet 模型之上，提出了一种轻量级全卷积的快速恶意域名检测模型 LW-CWNet。对于 CWNet 模型参数量及计算量较大的问题，本章提出的检测模型通过使用深度可分离卷积代替传统的卷积神经网络模型构建轻量级的并行卷积神经网络，能够更有效地处理输入的域名向量表示，以减少其存储大小，同时保持模型性能。全连接层可以使模型获得更高的准确率，但是高精度的代价与参数数量和计算速度相关，所以之后使用参数量少的全局平均池化层代替全连接层，减少训练参数数量、内存消耗和计算时间。对于模型泛化能力较差且模型轻量化带来的性能下降的问题，本章模型采用标签平滑方法防止模型在训练过程中过拟合，提高模型的泛化能力。本文实验部分评估了深度可分离卷积，全局平均池和标签平滑对性能和存储空间的影响。实验结果表明，所提出的 LW-CWNet 模型在参数数量显著减少及计算时间显著缩短的情况下有较好的检测性能，与其他恶意域名检测模型相比，在对域名分类的效率，精度和模型的大小上有显著提高。

4.2 模型结构

4.2.1 模型构造

如图 4.1 是所提出的基于全卷积的轻量级恶意域名检测检测模型 LW-CWNet。LW-CWNet 模型总体的结构由嵌入层，字符和词的深度可分离卷积层，轻量级全局平均池化层和标签平滑修正后的损失函数组成，对合法和恶意域名进行二分类。首先，对 CWNet 提取域名特征的卷积神经网络使用深度可分离卷积代替，其深度卷积和逐点卷积可以在得到相同特征图的条件下显著减少模型计算量，训练时长和参数数量[60]。之后，由于全连接层冗余参数过多，提出一种使用卷积和全局平均池化相结合的轻量级全局平均池化将其代替，减少大量模型参数。然后，通过使用标签平滑修正损失函数提高模型对标签的利用度，提升模型的鲁棒性。最终，得到一种计算量较少的、参数数量较小的、可实现较高分类精度的恶意域名检测模型，在恶意和合法域名的分类检测中表现出色[61-63]。

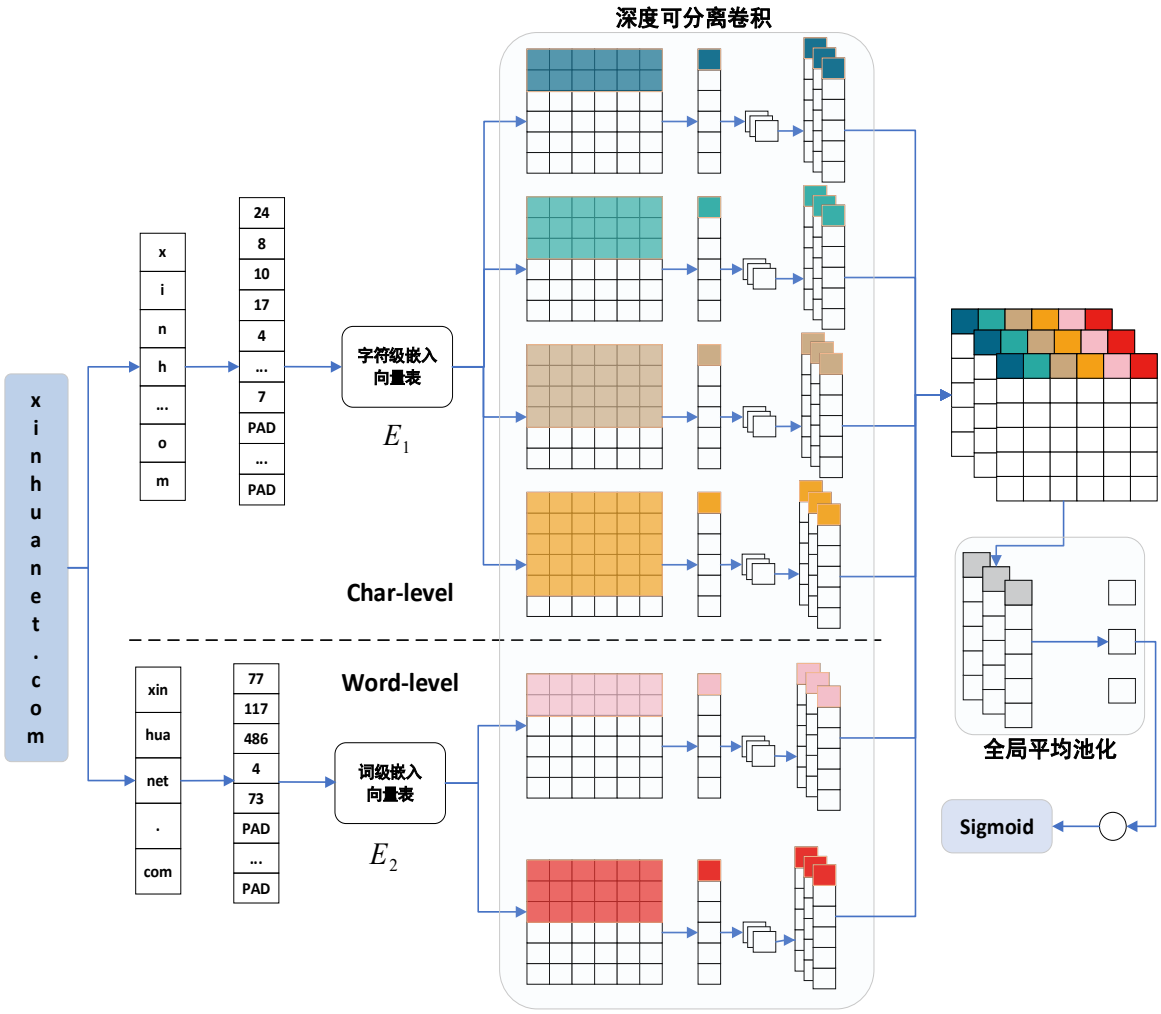

图 4.1 LW-CWNet 网络结构图

4.2.2 域名嵌入

域名不区分大小写，由多个字母、数字和特殊字符如连词符“-”，分隔符“.”连接而成。为了可以使用卷积神经网络对合法或恶意域名进行分类，需要将域名输入到嵌入层得到字符或单词的向量化表示。

字符级部分和词级部分都使用 Word2Vec 对域名进行向量化表示。将每个字符或词转化为长度为 D 的一维向量，D 设置为 30。数据集中最长域名的字符个数为 67，所以将字符级和词级向量表示的长度 L_1 和 L_2 都设置为 67，长度小于 67 的域名字符串使用零向量填充。利用该向量化方法将数据集中每条域名的单个字符或词转换为向量表示，拼接所有域名字符或词，获得整条域名大小为 30×67 的二维矩阵向量化表示，如图 4.2 所示。

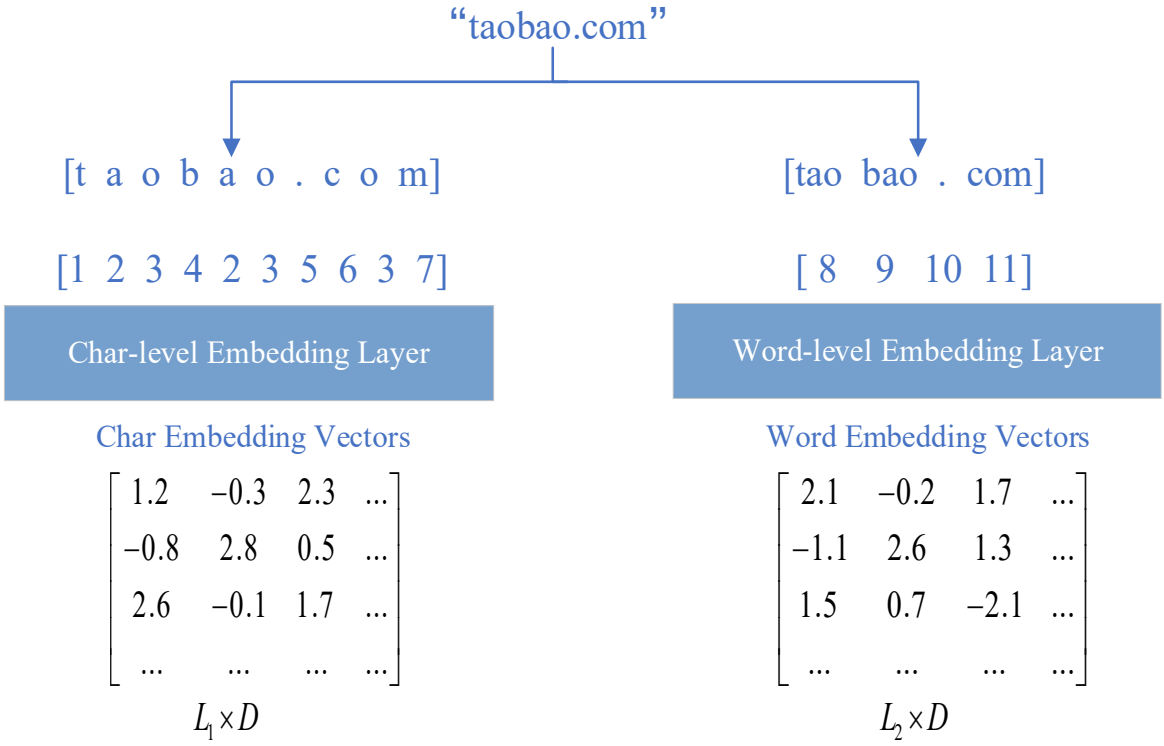

图 4.2 域名嵌入

4.2.3 并行的深度可分离卷积结构

得到域名的嵌入向量表示后，CWNet 采用包含不同卷积核大小的并行卷积网络模型对输入的域名进行特征提取，可以得到不同长度的字符或词之间相关的特征信息。模型共使用 6 个卷积分支来对域名进行特征提取，分别将域名嵌入表示输入到 4 个字符级卷积分支和 2 个词级卷积分支，得到 6 组具有不同卷积感知范围的字符及词的特征。使用这种方式可以有效地得到域名不同类型的特征，但是 6 个分支都分别使用 128 个相同大小的卷积核进行卷积计算，使得模型的参数量较多，计算量较大。

对 CWNet 模型使用深度可分离卷积来代替标准卷积，可以减少模型参数和计算量，实现轻量级的域名检测模型。深度可分卷积结构由深度卷积和逐点卷积两部分构成，组合提取域名的特征[64]。与标准卷积神经网络相比，深度可分离卷积可以在模型精度略微下降的情况下，大幅度减少模型参数数量和计算量。深度可分卷积结构首次提出于视觉领域，主要解决传统卷积神经网络内存需求和运算量大，无法在移动设备以及边缘设备上运行的问题。与视觉领域的卷积运算不同的是，使用卷积神经网络进行恶意域名检测其卷积核大小只与卷积核的长度 D_k 相关，卷积核的宽度等于域名嵌入表示长度 D，卷积核只进行纵向计算。为了可以使用深度可分卷积结构进行一维卷积运算，对其进行了改进如图 4.3 所示[65]。

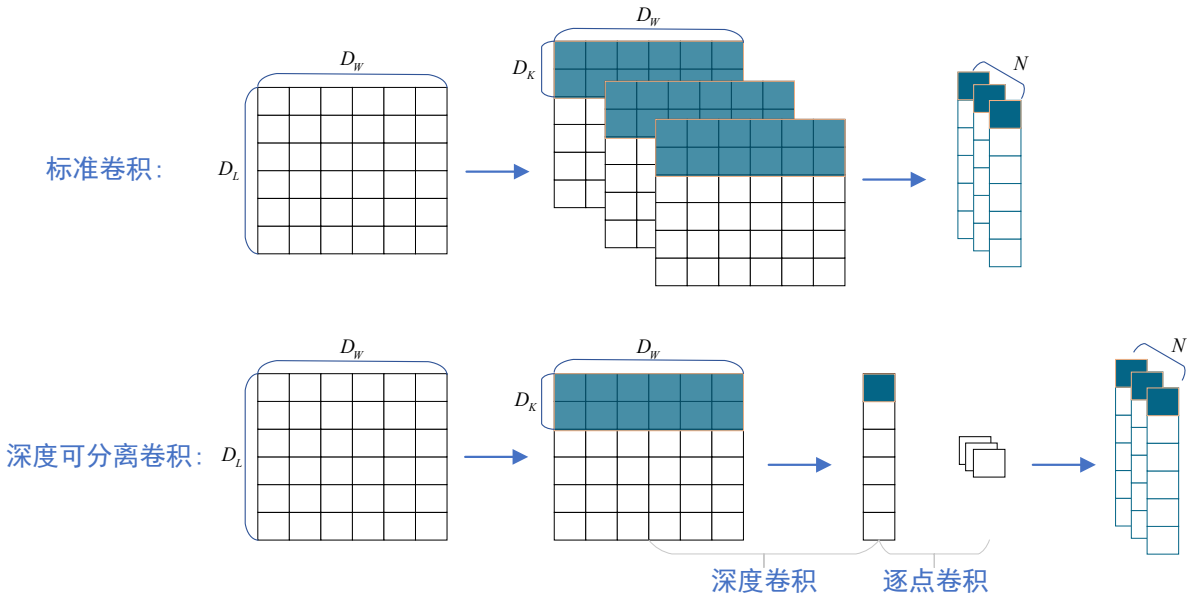

图 4.3 卷积方式对比图

在一维卷积神经网络中，D_L 是输入域名的字符长度，D_W 是输入域名字符的维度，D_K 是卷积核大小，N 是输出通道数。在使用一维卷积进行文本分类中，输入仅一个通道，尺寸大小为 $D_L \times D_W$。在标准卷积中，输入特征矩阵的通道数等于卷积核通道数，卷积核个数等于输出特征矩阵通道数，每个卷积分支使用 128 个卷积核对输入进行卷积运算得到 128 个 66×1 的特征图。对单个域名进行卷积的计算量为 $D_K \times D_W \times N \times D_L \times D_W$ [66]。深度可分离卷积结构使用深度和逐点卷积组合代替标准卷积，能够得到标准卷积大小相同的特征图。深度卷积的卷积核深度为 1，卷积核深度等于输入特征矩阵的深度也等于输出特征矩阵的深度。逐点卷积的卷积核大小为 1×1，其输入特征矩阵深度与卷积核深度相同，卷积核个数与输出特征的矩阵深度相同[67]。对单个域名进行卷积的计算量为 $D_K \times D_W \times D_L \times D_W + N \times D_L \times D_W$。因此，对于单个卷积分支，使用深度可分离卷积替换标准卷积后计算量比例为:

$$\frac{D_K \times D_W \times D_L \times D_W + N \times D_L \times D_W}{D_K \times D_W \times N \times D_L \times D_W} = \frac{1}{N} + \frac{1}{D_K \times D_W} \tag{4.1}$$

若 D_K 为 2，D_W 为 30，理论上标准卷积计算量大约是深度可分离卷积的 60 倍。通过使用深度可分离卷积，卷积模型的参数可以显著减少。

对于包含多个卷积分支的并行卷积架构，参数化简为:

$$\sum_{i=1}^{n} \frac{(D_k^n \times D_W \times D_L \times D_W + N \times D_L \times D_W)}{D_k^n \times D_W \times N \times D_L \times D_W} = \frac{n}{N} + \sum_{i=1}^{n} \frac{1}{D_k^n \times D_W} \tag{4.2}$$

D_k^n 是第 n 个卷积分支卷积核大小，所提出的模型 LW-CWNet 共使用 6 个卷积分支。其中有 4 个字符级卷积分支卷积核大小分别是 2，3，4，5，有 2 个词级卷积分支卷积核大小分别是 2，3。

4.2.4 轻量级全局平均池化

在使用神经网络模型对文本或者图片分类的早期，为了达到更好的分类精度，全连接层通常被作为最后几层，组合提取到的特征来最终确定输入是哪一个分类。但是全连接层冗余参数多，容易过拟合，而全局平均池化层能够有效地大幅减少模型参数[68]。本章节所提出轻量级模型中也将使用全局平均池化层。为了使并行卷积神经网络所提取的域名信息更好的被利用，使输出层维持更高的分类精度，提出一种轻量级的全局平均池化层。本章所提出的轻量级的全局平均池化层如图 4.4 所示。

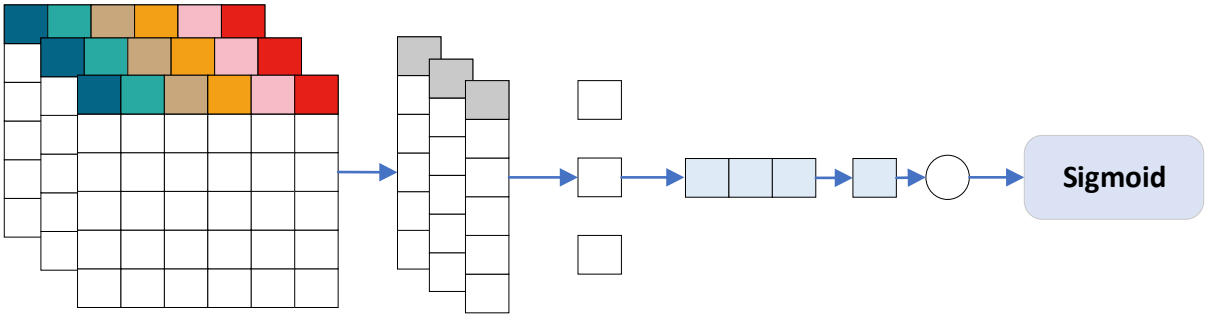

图 4.4 轻量级全局平均池化结构图

首先，将 2 个提取域名词级特征的并行卷积分支与 4 个提取域名字符级特征的并行卷积分支的输出进行拼接。其次，为了从字符和词特征融合后的 66×6 特征图中进一步提取深层特征，使用大小为 1 的卷积核再次进行卷积运算得到 128 个 66×1 的特征图。之后，对每个特征表示进行全局平均池化，可以减少大量冗余参数，得到融合字符和词特征的高效特征表示。最后，将多个融合特征进行拼接，拼接成一维特征向量，仅使用 1 个大小为 1 的卷积核对其进行卷积运算输出唯一的特征，使用 Sigmoid 函数得到最终域名合法或恶意的预测结果。所提出的轻量级全局平均池化层与全连接层相比，可以显著减少参数数量[69, 70]。

4.2.5 标签平滑

标签平滑是一种防止模型过拟合的正则化方法，是对损失函数的修正，用于减少模型对训练数据中噪声和标签错误的敏感性，常用于分类任务，通过在分类问题中的标签上引入噪声，可以帮助模型在训练过程中避免过拟合，同时也可以提高模型的鲁棒性和泛化能力。

标签平滑的基本思想是将正确的标签设置为一个略小的值，而将错误的标签设置为一个略大的值。缓解了二分类任务过于绝对的问题（非 1 即 0），可以增强模型的泛化性能。例如，对于域名的分类任务中，原本的标签假设恶意域名设置为 1 合法域名设置为 0。使用 Focal Loss 损失函数会使得在训练过程中对预测结果信任度很高，可能会降低模型的泛化能力。若是在大型数据集中包含标签错误的数据，神经网络模型应该对这个标签错误的数据保持怀疑，而不是完全相信，减少围绕错误数据的建模。如公式(4.3)所示，神经网络的训练目标在使用标签平滑方法后从“1”调整为“$1-\alpha$”，而不是简单设置为 1，α 为平滑因子，K 为类别数[71]。网络模型对经过标签平滑处理的数据信任度不高，因此能够对真实数据进行泛化且表现出色[72-74]。

$$y_c = \begin{cases} 1-\alpha & \text{if } c = label, \\ \alpha/(K-1) & otherwise. \end{cases} \tag{4.3}$$

如图 4.5 所示为不同 α 情况下的标签平滑在 β 为 1 的 Focal Loss 损失函数上的修正影响。黑色虚线表示标准的 Focal Loss 损失函数，其他线条为标签平滑影响后的效果。在本章 α 设置为 0.1，K 设置为 2。

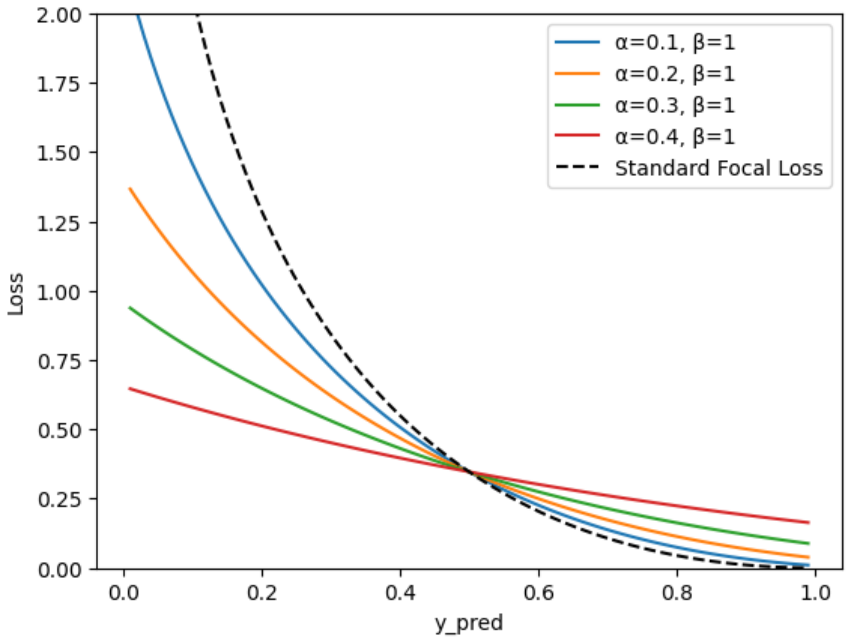

图 4.5 标签平滑修正损失函数效果图

4.3 实验结果及分析

4.3.1 数据集

为了对提出的轻量级恶意域名检测模型进行全面的评估，使用两种不同的数据集对模型进行测试。第一类数据集（DS1）是对上一章节数据集的扩充。该数据集存在多个数量较少的 DGA 家族，如 blackhole，ccleaner，madax 等仅有个位数的数据，较难检测的 matsnu 家族域名不足 1000 条。为了使恶意域名检测模型学习到更多的域名特征，提升模型的检测能力，使用在文献[75]中找到域名生成算法生成对应的恶意域名对数据集进行扩充，将部分数量较少的 DGA 域名扩充到 10000 条[75]。DS1 数据集中包括 56 个 DGA 家族的恶意域名共 578647 条，且选取 Alexa 网站前 580000 条域名作为合法域名。并将所有数据按照通 4:1 的比例划分训练集、测试集。第二类数据集（DS2）来源于文献[76]中 AmritaDGA 数据集[76]。AKAMAI 公司在处理 DNS 查询中发现解析的所有域名中恶意域名约占总域名数量的 20%。为了模拟现实域名的解析环境，按照合法和恶意样本的 4:1 比例构建测试集，其中包括 80000 条合法域名和 20000 条恶意域名。数据集统计如表 4.1 所示。

表 4.1 两种域名数据集描述

	类别	合法域名	恶意域名	总数
DS1	训练集	464000	462918	926918
	测试集	116000	115729	231729
DS2	测试集	80000	20000	100000

4.3.2 实验环境及评价指标

本章节实验的实验环境如表 4.2 所示。本文实验的训练参数设置如下：训练周期（Epochs）为 30，批处理量（BatchSize）为 256，使用 Adam 优化器，初始学习率设置为 0.001。

表 4.2 第 4 章实验环境表

实验环境	参数
CPU 处理器	AMD Ryzen 7 5800H 3.30GHz
GPU	NVIDIA GeForce RTX 3060 6GB
内存	16GB
操作系统	Windows11
加速库	CUDA11.0
深度学习框架	Pytorch1.7.1
编程工具	Pycharm

设定恶意域名的标签为 1，合法域名的标签为 0。本章节提出的是轻量级检测模型，但是检测精度仍是首要考虑的问题，其次是减少模型的参数，降低占用空间，提高对待测域名的检测速度。为了综合评估模型的检测效果，实验采用的评价标准包括准确率、召回率、精确率、F1 值、模型占用内存大小、CPU 处理单个域名所需的时间和 AUC 七个指标。

4.3.3 对比实验

为了更综合地验证在 CWNet 基础上提出的 LW-CWNet 模型的有效性，在 DS1 和 DS2 数据集上与其他 4 种分类模型和 CWNet 模型进行了对比实验，如表 4.3 所示。

表 4.3 五种对比模型

模型结构	参考文献或章节
Bi-LSTM	[44]
SVDCNN	[45]
TextCNN	[47]
LW-TextCNN	[48]
CWNet	本文第三章所提模型

（1）DS1 数据集对比实验

从表 4.4 和图 4.6 中可以明显看出本文第三章提出的基于字符和词特征融合的恶意域名检测模型 CWNet 在准确率、召回率、精确率、F1 值和 AUC 上均表现非常出色，但是相对于其他模型，在提取域名字符特征使用了 4 个并行的卷积分支，在提取域名词级特征使用了 2 个并行的卷积分支使得模型参数过多，占用内存较大，对单个域名进行推理的时间也更多。LW-CWNet 是一个轻量级域名检测模型，表现也很不错。虽然相对于 CWNet 模型，准确率、召回率、精确率和 F1 值都有所降低，但是使用深度可分离卷积和轻量级全局平均池化进行改进后，该模型的模型大小和对单个域名推理的时长也大大减小。Bi-LSTM 是一个双向长短期记忆网络模型，相对于其他模型，模型精度略低一些，但是该模型结构简单，在模型大小及推理时长上表现优异。SVDCNN 是一个深层的卷积模型，该模型相对较大，推理时间也较长。TextCNN 由于也是使用并行的卷积分支，在各个指标上都有较好的表现。相对于其他模型，模型较大，推理时长较短。LW-TextCNN 是对 TextCNN 的轻量级模型，检测精度不如 TextCNN，但是该模型比 TextCNN 小，推理所需时长也更短。结果表明，所提出的轻量化域名检测模型 LW-CWNet 在检测精度上有所下降，但是模型减少了大量参数数量，使得模型更小，推理时长更少，综合表现优秀。

表 4.4 DS1 数据集的对比实验结果

检测模型	准确率	召回率	精确率	F1 值	模型大小	时长
Bi-LSTM	0.9655	0.9530	0.9772	0.9649	7.6MB	1.6ms
SVDCNN	0.9706	0.9592	0.9811	0.9700	22.3MB	4.2ms
TextCNN	0.9782	0.9729	0.9801	0.9765	27.8MB	2.6ms
LW-TextCNN	0.9752	0.9721	0.9778	0.9749	12.4MB	**1.2ms**
CWNet	**0.9853**	**0.9817**	**0.9879**	**0.9848**	35.6MB	4.7ms
LW-CWNet	0.9794	0.9754	0.9832	0.9793	**4.6MB**	1.4ms

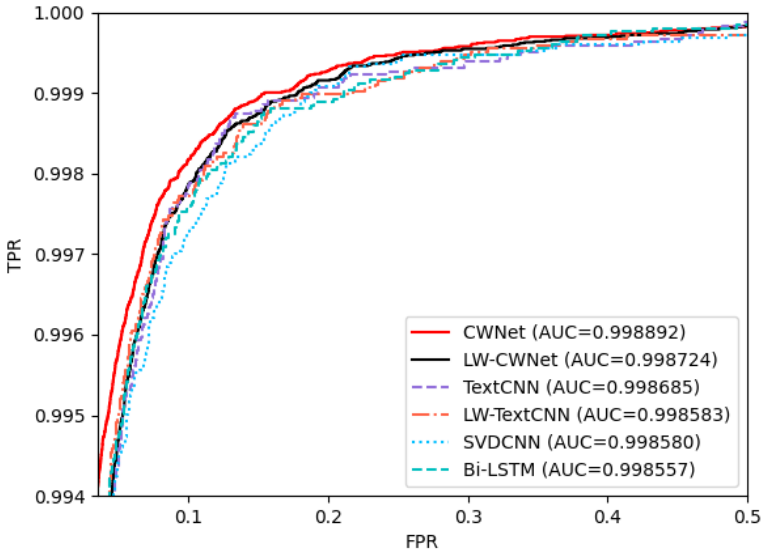

图 4.6 DS1 数据集的对比模型 AUC

（2）DS2 数据集对比实验

如表 4.5 所示，为了进一步确认所提出的轻量级检测模型 LW-CWNet 在真实环境下的有效性，使用 DS2 数据集对模型进行测试。CWNet 在 DS2 数据集上与其他模型相比同样表现出最好的检测精度，在准确率、召回率、精确率、F1 值上均明显高于对比模型。LW-CWNet 模型的检测准确率为 0.9692 仅略差于 CWNet。

表 4.5 DS2 数据集的对比实验结果

检测模型	准确率	召回率	精确率	F1 值	模型大小	推理时间
Bi-LSTM	0.9553	0.9411	0.9632	0.9520	7.6MB	1.6ms
SVDCNN	0.9577	0.9479	0.9665	0.9571	22.3MB	4.7ms
TextCNN	0.9659	0.9507	0.9723	0.9614	27.8MB	2.6ms
LW-TextCNN	0.9642	0.9501	0.9711	0.9605	12.4MB	**1.2ms**
CWNet	**0.9703**	**0.9596**	**0.9808**	**0.9701**	35.6MB	5.2ms
LW-CWNet	0.9692	0.9563	0.9773	0.9667	**4.6MB**	1.4ms

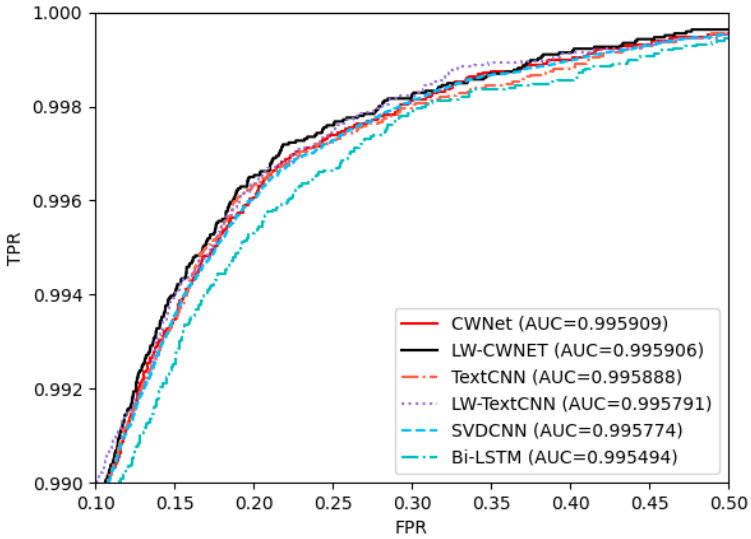

图 4.7 DS2 数据集的对比模型 AUC

如图 4.7 所示，在 FPR 为 0.1 到 0.5 之间 LW-CWNet 的 AUC 高于其他模型，但是整体的 AUC 不如 CWNet。其他模型在 DS2 数据集上也表现良好，但是在召回率上有明显的下降，不如 LW-CWNet 稳定。结果表明，所提出的 LW-CWNet 模型在真实条件下仍然在检测精度和稳定性方面都有较好的表现。

在 DS1 和 DS2 数据集上的对比实验结果表明，本章在 CWNet 基础上提出的 LW-CWNet 模型的总体检测效果优于其他对比模型。该模型可以提取到域名更多的特征，有较好的检测精度，同时网络模型进行了轻量化，在模型大小及推理速度上也优于其他模型。提出的 LW-CWNet 模型在检测精度、模型大小及计算速度之间取得了较好的平衡。

4.3.4 消融实验

本章模型 LW-CWNet 是在 CWNet 模型的架构上进行的轻量化，首先使用轻量级的全局平均池化代替全连接层得到 CWNet+LGAP 架构进行实验，然后使用并行的深度可分离卷积代替传统卷积得到 DS-CWNet+LGAP 架构进行实验，最后使用标签平滑修正损失函数得到 DS-CWNet+LGAP+LS 架构，也就是整体的 LW-CWNet 进行实验。实验数据如表 4.6 所示，不同架构的 AUC 如图 4.8 所示。

对模型所提出的具体改进是为了减少模型参数，实现模型轻量化。CWNet 模型在准确率、召回率、精确率、F1 值上均表现最好，但是模型太大占用 35.6MB 的内存空间。由于全连接层的参数量占比最高，因此首先使用轻量级全局平均池化 LGAP 代替。使用 LGAP 后，检测效果有所下降，但模型参数明显减少，模型从 35.6M 下降到 6.7M。为了进一步减少模型参数，使用 DS-CWNet 代替 CWNet，模型从 6.7M 减小到 4.6M，模型的检测精度再次下降。为了提高模型的检测精度

且不增加模型的参数量，对损失函数使用标签平滑进行修正，模型大小保持不变且检测精度略有提升，模型的 AUC 增加到 0.99872，使用标签平滑可以在不增加模型参数的情况下提高模型的检测精度。

表 4.6 DS1 数据集的消融实验结果

模型	准确率	召回率	精确率	F1 值	模型大小
CWNet	**0.9853**	**0.9817**	**0.9879**	**0.9848**	35.6MB
CWNet+LGAP	0.9834	0.9774	0.9842	0.9808	6.7MB
DS-CWNet+LGAP	0.9782	0.9732	0.9816	0.9774	4.6MB
DS-CWNet+LGAP+LS	0.9794	0.9754	0.9832	0.9793	**4.6MB**

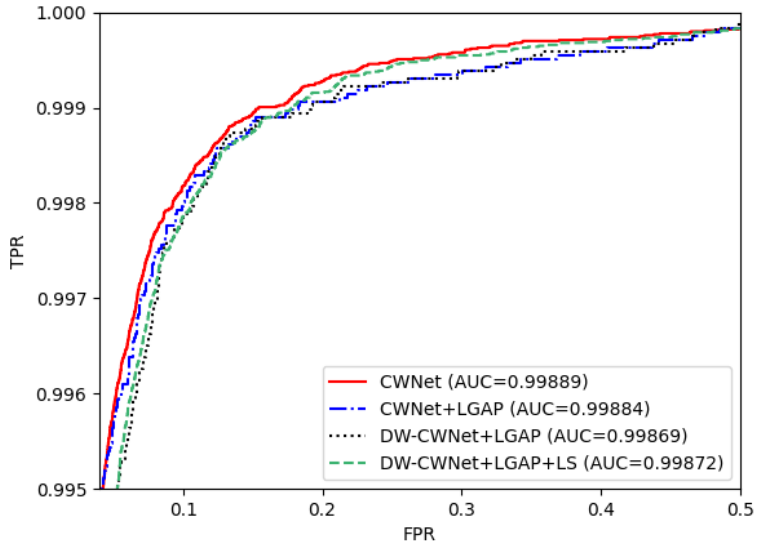

图 4.8 DS1 数据集的消融实验 AUC

消融实验表明，所提出的 DS-CWNet 和 LGAP 可以在略微降低模型检测性能的情况下，显著减少模型大小。使用的标签平滑可以在不增加模型参数的情况下有效提高模型的检测性能。LW-CWNet 模型在只保留 CWNet 模型 1/8 左右参数量的情况下，仍能保持较高的检测性能。

通过一系列实验证明，本章提出的 LW-CWNet 模型，其检测精度表现优秀，且模型参数更少，推理速度更快，实现了检测精度和计算速度的平衡。LW-CWNet 模型具有更好的实际应用价值，在现实场景中能够更加高效地进行恶意域名检测。

4.4 本章小结

本章提出了一种轻量级的快速恶意域名检测模型 LW-CWNet。所提出的轻量级域名检测模型由并行深度可分离卷积网络 DS-CWNet 和轻量级全局平均池化 LGAP 结构组成，DS-CWNet 可快速提取域名中不同长度的字符及词之间的关系特征，LGAP 可以替代参数较多的全连接层，快速从拼接后的特征图中得到全局

特征。之后使用标签平滑修正损失函数，提高模型鲁棒性。为了综合评估所提出的模型，在扩充的 DS1 数据集和模拟真实条件下设定的 DS2 数据集上进行实验。实验结果表明，所提模型 LW-CWNet 在参数数量显著减少及计算时间显著缩短的情况下，有较好的检测性能。

总结与展望

近年来，随着互联网的普及，恶意域名检测成为网络安全领域的一个重要研究方向。恶意域名的检测可以帮助人们识别和阻止网络攻击、恶意软件的传播和其他网络安全威胁。在恶意域名检测研究起步期，研究者主要采用传统的机器学习算法来实现该任务，通过手工设计特征和分类器进行分类。但是这些传统方法面对越来越复杂的网络安全攻击时，效果受到限制。深度学习技术发展快速，使用深度学习方法来进行恶意域名检测逐渐成为主流方法，例如卷积神经网络、循环神经网络和生成对抗网络等。深度学习方法能够自动地学习特征，提高了检测的准确率和鲁棒性。此外，也有研究者将深度学习和传统机器学习算法相结合，取得了更好的效果。在恶意域名检测领域，公开的数据集如 Majestic Million、Alexa Top 1M、DGA 和 Malware Domain List 等被广泛应用于算法的训练和测试，恶意域名检测的准确率和鲁棒性得到了显著提高。未来，深度学习技术将在恶意域名检测领域发挥越来越重要的作用，为网络安全保驾护航。

基于深度学习的恶意域名检测研究已经取得了一定的进展，但还有很多问题需要解决。本文提出的 CWNet 可以提取到域名中字符和词的融合特征信息，在恶意域名检测领域取得了不错的效果。但是该网络模型训练参数过多，难以在现实场景中使用，之后在该网络模型的基础之上构建轻量化模型 LW-CWNet，可以在参数数量显著减少及计算时间显著缩短的情况下，有较好的检测性能。

（1）在恶意域名检测中，现有的模型仅提取域名的单一特征忽略了字符和词之间的关联语义，对于由随机字符组成的恶意域名有较好的检测效果，但是对由随机单词组成的 DGA 域名的检测性能不佳。为了提高模型对任何类型域名的检测能力，首先对待测域名进行向量化；之后使用并行卷积神经网络分别提取域名中字符和词的特征，对域名字符串所提供的信息利用度更高；最后将这两种特征进行拼接构造成融合特征，使用 Softmax 函数实现域名的分类检测。Focal Loss 损失函数可以更加关注难分类的样本，减少简单样本对模型检测能力的影响，提高模型的分类性能。该模型能够提升对恶意域名的检测能力，尤其是对于更具挑战性的恶意域名家族的检测准确率提升效果更为明显。最后进行了大量实验，验证了该方法的有效性。

（2）CWNet 网络模型可以提供良好的准确性，但是由于大量可训练的参数导致较高的内存消耗，应用难度较大。为了解决这一问题，在该模型的基础上进行轻量化改进构建 LW-CWNet。使用深度可分离卷积代替传统卷积神经网络，可以更有效地处理输入的域名向量表示，同时减少参数量和计算量。使用全局平均

池化层代替全连接层，进一步减少参数数量和计算时间。采用标签平滑方法防止模型在训练过程中过拟合，提高模型的泛化能力。在两种数据集上实验的结果表明 LW-CWNet 模型在精度、效率和模型大小方面优于其他恶意域名检测模型，与 CWNet 模型相比，在参数数量显著减少及计算时间显著缩短的情况下，有较好的检测性能。

本文设计的模型在数据集上有着较好的分类效果，但仍存在许多不足，今后从以下几个方面进一步完善与研究。

（1）设计的模型主要是对待测域名进行合法或恶意的二分类，没有对恶意域名进行不同 DGA 家族的多分类。接下来重点研究目标是对恶意域名进一步的家族多分类检测。

（2）恶意域名的组成方式将会更加多样化。本文所提出的两种域名检测模型仅对现有的域名有较好的分类效果，为了应对以后未知的变种域名，需要持续关注恶意域名的相关信息，针对不同种类的恶意域名对模型进行相应的改进。

（3）模型仅在搜集的域名数据集上表现出色，尚未对现实网络环境的域名进行验证，之后将会把轻量级域名检测模型部署到系统用于真实的网络环境当中。

参考文献

[1] Singh M, Singh M, Kaur S. Issues and challenges in DNS based botnet detection: A survey [J]. Computers & Security, 2019, 86: 28-52.

[2] 杨宏宇, 那玉琢. 一种 Android 恶意软件检测模型 [J]. 西安电子科技大学学报, 2019, 46(3): 45-51.

[3] 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.

[4] Kaloudi N, Li J. The AI-Based Cyber Threat Landscape: A Survey [J]. ACM Computing Surveys, 2020, 53(1): 1-34.

[5] 王媛媛, 吴春江, 刘启和, 等. 恶意域名检测研究与应用综述 [J]. 计算机应用与软件, 2019, 36(9): 310-316.

[6] 赵宏, 常兆斌, 王乐. 基于词法特征的恶意域名快速检测算法 [J]. 计算机应用, 2019, 39(1): 227-231.

[7] Hong Z, Zhaobin C, Bao G, et al. Malicious Domain Names Detection Algorithm Based on N-Gram [J]. Journal of Computer Networks and Communications, 2019, (2): 1-9.

[8] Zhao H, Chang Z, Wang W, et al. Malicious Domain Names Detection Algorithm Based on Lexical Analysis and Feature Quantification [J]. Ieee Access, 2019, 7(9): 128990-128999.

[9] Saiyod S, Chanthakoummane Y, Benjamas N, et al. Improving intrusion detection on snort rules for botnet detection [J]. Software Networking, 2018, 2018(1): 191-212.

[10] Shi Y, Chen G, Li J. Malicious Domain Name Detection Based on Extreme Machine Learning [J]. Neural Processing Letters, 2018, 48: 1347-1357.

[11] Almashhadani A O, Kaiiali M, Carlin D, et al. MaldomDetector: A System for Detecting Algorithmically Generated Domain Names with Machine Learning [J]. Computers & Security, 2020, 93: 101787-101793.

[12] 马栋林, 张澍寰, 赵宏. 改进 Relief-C5.0 的恶意域名检测算法 [J]. 计算机工程与应用, 2022, 58(11): 100-106.

[13] Yang L, Liu G, Zhai J, et al. A novel detection method for word-based DGA[C]. 4th International Conference on Cloud Computing andSecurity, 2018: 472-483.

[14] Cucchiarelli A, Morbidoni C, Spalazzi L, et al. Algorithmically generated malicious domain names detection based on n-grams features [J]. Expert Systems with Applications, 2021, 170: 114551-114567.

[15] Alshdadi A A, Alghamdi A S, Daud A, et al. Blog backlinks malicious domain name detection via supervised learning [J]. International Journal on Semantic Web and Information Systems (IJSWIS), 2021, 17(3): 1-17.

[16] Cheng Y, Chai T, Zhang Z, et al. Detecting malicious domain names with abnormal whois records using feature-based rules [J]. The Computer Journal, 2022, 65(9): 2262-2275.

[17] 张斌, 廖仁杰. 基于关联信息提取的恶意域名检测方法 [J]. 通信学报, 2021, 42(10): 162-172.

[18] Yang L, Liu G, Liu W, et al. Detecting Multielement Algorithmically Generated Domain Names Based on Adaptive Embedding Model [J]. Security and Communication Networks, 2021, 2021: 1-20.

[19] Palaniappan G, Sangeetha S, Rajendran B, et al. Malicious domain detection using machine learning on domain name features, host-based features and web-based features [J]. Procedia Computer Science, 2020, 171: 654-661.

[20] Yang L, Zhai J, Liu W, et al. Detecting Word-Based Algorithmically Generated Domains Using Semantic Analysis [J]. Symmetry, 2019, 11(2): 176-190.

[21] Palaniappan G, Sangeetha S, Rajendran B, et al. Malicious Domain Detection Using Machine Learning On Domain Name Features, Host-Based Features and Web-Based Features [J]. Procedia Computer Science, 2020, 171: 654-661.

[22] Chen Y, Zhang S, Liu J, et al. Towards a deep learning approach for detecting malicious domains[C]. 2018 IEEE International Conference on Smart Cloud (SmartCloud), 2018: 190-195.

[23] 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.

[24] 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 (ICAIBD), 2018: 228-234.

[25] 袁辰. 基于对抗模型的恶意域名检测方法的研究与实现 [D]. 北京建筑大学, 2018.

[26] 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.

[27] Yanchen Q, Zhang B, Zhang W, et al. DGA Domain Name Classification Method Based on Long Short-Term Memory with Attention Mechanism [J]. Applied Sciences, 2019, 9(20): 4205-4221.

[28] Lai S, Xu L, Liu K, et al. Recurrent convolutional neural networks for text classification[C]. Proceedings of the AAAI conference on artificial intelligence, 2015, 29(1): 128-136.

[29] Zago M, Pérez M G, Pérez G M. Scalable detection of botnets based on DGA Efficient feature discovery process in machine learning techniques[C]. Soft Computing, 2020, 24(8): 5517-5537.

[30] 陈立皇, 程华, 房一泉. 基于注意力机制的 DGA 域名检测算法 [J]. 华东理工大学学报(自然科学版), 2019, 45(3): 478-485.

[31] Yang L, Liu G, Dai Y, et al. Detecting stealthy domain generation algorithms using heterogeneous deep neural network framework [J]. IEEE Access, 2020, 8: 82876-82889.

[32] Hwang C, Kim H, Lee H, et al. Effective DGA-Domain Detection and Classification with TextCNN and Additional Features [J]. Electronics, 2020, 9(7): 1070-1091.

[33] 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, 2020, 15(2): 2225-2240.

[34] 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.

[35] 张斌, 廖仁杰. 基于 CNN 与 LSTM 相结合的恶意域名检测模型 [J]. 电子与信息学报, 2021, 43(10): 2944-2951.

[36] 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), 2018: 1-8.

[37] 杨路辉, 刘光杰, 翟江涛, 等. 一种改进的卷积神经网络恶意域名检测算法 [J]. 西安电子科技大学学报, 2020, 47(1): 37-43.

[38] 王志强, 李舒豪, 池亚平, 等. 基于深度学习的恶意 DGA 域名检测 [J]. 计算机工程与设计, 2021, 42(3): 601-606.

[39] 李晓冬, 李育强, 宋元凤. 新的基于融合向量的 DGA 域名检测方法 [J]. 计算机应用研究, 2022, 39(6): 1834-1837+1844.

[40] Jiang Y, Jia M, Zhang B, et al. Malicious Domain Name Detection Model Based on CNN-GRU-Attention[C]. 33rd Chinese Control and Decision Conference, 2021: 1602-1607.

[41] Berman D. DGA CapsNet: 1D application of capsule networks to DGA detection [J]. Information, 2019, 10(5): 157-169.

[42] Yang L, Liu G, Wang J, et al. Fast3DS: A real-time full-convolutional malicious domain name detection system [J]. Journal of Information Security and Applications, 2021, 61: 102933-102952.

[43] Aarthi B, Jeenath Shafana N, Flavia J, et al. A Hybrid Multiclass Classifier Approach for the Detection of Malicious Domain Names Using RNN Model [M]. Computational Vision and Bio-Inspired Computing: Proceedings of ICCVBIC 2021. Springer. 2022: 471-482.

[44] Namgung J, Son S, Moon Y-S. Efficient deep learning models for dga domain detection [J]. Security and Communication Networks, 2021, 2021: 1-15.

[45] Singh A, Roy P K. Malicious URL detection using multilayer CNN[C]. 2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), 2021: 340-345.

[46] Solovyeva E, Abdullah A. Binary and Multiclass Text Classification by Means of Separable Convolutional Neural Network [J]. Inventions, 2021, 6(4): 70-82.

[47] Huang X, Li H, Liu J, et al. A Malicious Domain Detection Model Based on Improved Deep Learning [J]. Computational Intelligence and Neuroscience, 2022, 2022: 26-42.

[48] Yadav R. Light-weighted CNN for text classification [J]. arXiv preprint arXiv:2004.07922, 2020.

[49] Satoh A, Nakamura Y, Nobayashi D, et al. Estimating the Randomness of Domain Names for DGA Bot Callbacks [J]. IEEE Communications Letters, 2018, 22(7): 1378-1381.

[50] Choudhary C, Sivaguru R, Pereira M, et al. Algorithmically generated domain detection and malware family classification[C]. International Symposium on Security in Computing and Communication, 2018: 640-655.

[51] 方圆, 李明, 王萍, 等. 基于混合卷积神经网络和循环神经网络的入侵检测模型 [J]. 计算机应用, 2018, 38(10): 2903-2907+2917.

[52] Johnson R, Zhang T. Effective use of word order for text categorization with convolutional neural networks [J]. arXiv preprint arXiv:1412.1058, 2014.

[53] Koroniotis N, Moustafa N, Sitnikova E. Forensics and Deep Learning

Mechanisms for Botnets in Internet of Things: A Survey of Challenges and Solutions [J]. IEEE Access, 2019, 7: 61764-61785.

[54] Shibahara T, Yamanishi K, Takata Y, et al. Malicious URL sequence detection using event de-noising convolutional neural network[C]. 2017 IEEE International Conference on Communications (ICC), 2017: 1-7.

[55] Le H, Pham Q, Sahoo D, et al. URLNet: Learning a URL representation with deep learning for malicious URL detection [J]. arXiv preprint arXiv:180203162, 2018.

[56] 杜鹏, 丁世飞. 基于混合词向量深度学习模型的 DGA 域名检测方法 [J]. 计算机研究与发展, 2020, 57(2): 433-446.

[57] Abdi F D, Wenjuan L. Malicious URL detection using convolutional neural network [J]. Journal International Journal of Computer Science, Engineering and Information Technology, 2017, 7(6): 1-8.

[58] 彭曦晨, 葛斌, 郃悦. 基于特征融合和注意力的图像分类研究 [J]. 合肥学院学报(综合版), 2022, 39(2): 91-97.

[59] Lin T-Y, Goyal P, Girshick R, et al. Focal Loss for Dense Object Detection [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 318-327.

[60] Jacovi A, Shalom O S, Goldberg Y. Understanding convolutional neural networks for text classification [J]. arXiv preprint arXiv:1809.08037, 2018.

[61] 崔甲, 施蕾, 李娟, 等. 一种高效的恶意域名检测框架 [J]. 北京理工大学学报, 2019, 39(1): 64-67.

[62] 刘敬学, 孟凡荣, 周勇, 等. 字符级卷积神经网络短文本分类算法 [J]. 计算机工程与应用, 2019, 55(5): 135-142.

[63] Lee J Y, Dernoncourt F. Sequential short-text classification with recurrent and convolutional neural networks [J]. arXiv preprint arXiv:1603.03827, 2016.

[64] Wang J, Wang Z, Zhang D, et al. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification[C]. IJCAI, 2017: 3172077-3172295.

[65] 张曼, 夏战国, 刘兵, 等. 全卷积神经网络的字符级文本分类方法 [J]. 计算机工程与应用, 2020, 56(5): 166-172.

[66] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications [J]. arXiv preprint arXiv:1704.04861, 2017.

[67] 杨路辉, 白惠文, 刘光杰, 等. 基于可分离卷积的轻量级恶意域名检测模型

[J]. 网络与信息安全学报, 2020, 6(6): 112-120.

[68] Taha I B, Ibrahim S, Mansour D-E A. Power transformer fault diagnosis based on DGA using a convolutional neural network with noise in measurements [J]. IEEE Access, 2021, 9: 111162-111170.

[69] Hsiao T-Y, Chang Y-C, Chou H-H, et al. Filter-based deep-compression with global average pooling for convolutional networks [J]. Journal of Systems Architecture, 2019, 95: 9-18.

[70] Li Z, Wang S H, Fan R R, et al. Teeth category classification via seven‐layer deep convolutional neural network with max pooling and global average pooling [J]. International Journal of Imaging Systems and Technology, 2019, 29(4): 577-583.

[71] Guo B, Han S, Han X, et al. Label confusion learning to enhance text classification models[C]. Proceedings of the AAAI conference on artificial intelligence, 2021: 12929-12936.

[72] Lukasik M, Bhojanapalli S, Menon A, et al. Does label smoothing mitigate label noise?[C]. International Conference on Machine Learning, 2020: 6448-6458.

[73] Lienen J, Hüllermeier E. From label smoothing to label relaxation[C]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021: 8583-8591.

[74] Müller R, Kornblith S, Hinton G E. When does label smoothing help? [J]. Advances in neural information processing systems, 2019, 32.

[75] 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.

[76] Zago M, Pérez M G, Pérez G M. UMUDGA: A dataset for profiling DGA-based botnet [J]. Computers & Security, 2020, 92: 101719-101733.