兰州理工大学

硕士论文

学校代号 10731　　学　　号 212081203037

分 类 号 TP391　　密　　级 公开

兰州理工大学

LANZHOU UNIVERSITY OF TECHNOLOGY

硕士学位论文

融合多类型特征的恶意域名检测研究

学位申请人姓名 韩力毅

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 计算机应用技术

研 究 方 向 模式识别与人工智能

论文提交日期 2024 年 4 月 10 日

学校代号：10731

学　　号：212081203037

密　　级：公　开

兰州理工大学硕士学位论文

融合多类型特征的恶意域名检测研究

学位申请人姓名：　韩力毅

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：　计算机应用技术

论文提交日期：　2024 年 4 月 10 日

论文答辩日期：　2024 年 5 月 20 日

答辩委员会主席：　火久元 教授

Research on Malicious Domain Name Detection Based on Multi-type Feature Fusion

by

HAN Liyi

B.E. (Lanzhou University of Technology) 2021

A thesis submitted in partial satisfaction of the

Requirements for the degree of

Master of Professional

in

Computer Application Technology

in the

School of Computer and Communication

Lanzhou University of Technology

Supervisor

Professor ZHAO Hong

May, 2024

目　录

摘要

随着互联网的迅速普及，各种网络攻击事件频发，包含恶意链接和流量攻击等。恶意域名作为一种网络攻击方法，经常被用于垃圾邮件传播、网络钓鱼攻击、恶意软件分发以及为僵尸网络托管命令与控制服务器等恶意活动。因此，及时检测并防范恶意域名引起的网络攻击显得尤为重要，这不仅是维护互联网正常运行的必要措施，更是保护用户免受网络攻击的重要保障。

现有恶意域名检测模型大多采用深度学习方法，效果较好，但仍存在部分亟待解决的问题，主要集中在：1）依赖于单一类型特征，域名特征的多样性考虑不足；2）域名潜在特征提取不充分，导致部分构词复杂的恶意域名检测效果不佳；3）泛化性不足，难以适应不断变化的恶意域名。本文针对以上问题展开研究，主要研究内容如下。

（1）鉴于现有方法存在捕获域名潜在特征不充分、域名特征的多样性考虑不足、泛化性不足的问题。提出一种融合域名文本和注册特征的恶意域名检测方法，首先，从公开渠道收集合法和恶意域名，并获取域名 WHOIS 注册信息；其次，结合 Transformer 和卷积神经网络(Convolutional Neural Network, CNN)模型提取域名全局和局部特征；最后，融合两种特征，从而实现域名的分类。在新冠疫情相关数据集和公开数据集 CICBellDNS2021 上进行了实验，结果表明，域名注册特征能够有效提高恶意域名检测效果，相较于对比模型，准确率和精确率分别提升了 0.54%和 1.08%，此外，对于不同类型数据集泛化性较强。

（2）融合域名解析特征丰富域名语义表达，以实现恶意域名分类。首先，从公开渠道收集合法和恶意域名，并获取域名 WHOIS 注册信息和域名系统(Domain Name System, DNS)解析记录；其次，采用基于注意力机制的特征融合模块融合域名文本、注册和解析特征；最后，设计了一种恶意域名检测模型 Iconformer，结合注意力机制的全局建模和 CNN 的局部建模优点，并引入马卡龙式前馈模块充分捕获域名多级特征。在合法域名数据集 Cisco Umbrella 和 CICBellDNS2021 等九个恶意域名数据集上进行了对比、消融和泛化性实验，结果表明，该方法能显著提高恶意域名检测效果和泛化性，相较于对比模型，准确率和精确率分别提升了 1.17%和 0.94%。

关键词：恶意域名检测；多类型特征；特征融合；WHOIS 注册信息；DNS 解析记录

Abstract

With the rapid popularization of the internet, various network attack incidents, including malicious links and traffic attacks, are becoming more frequent. Malicious domain names are often used as a method of network attacks, commonly found in spam email propagation, phishing attacks, malicious software distribution, and hosting command and control servers for botnets. Therefore, timely detection and prevention of network attacks caused by malicious domain names are crucial, this is not only a necessary measure to maintain the normal operation of the internet but also a crucial guarantee to protect users from cyber attacks.

Existing malicious domain names detection models mostly use deep learning methods, which have shown good effectiveness but still have some urgent problems to be resolved, mainly focusing on: 1) relying on a single type of feature, lacking consideration for the diversity of domain name features; 2) insufficient extraction of potential domain name features, resulting in poor detection performance for complexly constructed malicious domain names; 3) lack of generalization, making it challenging to adapt to constantly evolving malicious domain names. This thesis focuses on researching the above problems, with the main research content as follows.

(1) Considering the shortcomings of existing methods in insufficiently capturing domain name features, lack of diversity in domain name features, and inadequate generalization, a method for detecting malicious domain names is proposed, which integrates domain text and registration features. Firstly, benign and malicious domain names are collected from public sources, and obtain the WHOIS registration information. Secondly, global and local domain name features are extracted using a combination of Transformer and Convolutional Neural Network (CNN) models. Finally, the two types of features are fused to achieve domain classification. Experiments conducted on the COVID-19-related dataset and the public dataset CICBellDNS2021 demonstrate that registration features can effectively improve the accuracy of malicious domain names detection. Compared with the comparative models, the accuracy and precision are improved by 0.54% and 1.08%, respectively, and the model also

demonstrates strong generalization performance across different types of datasets.

(2) Enriching the semantic expression of domain names by incorporating domain name resolution features to achieve malicious domain name classification. Firstly, benign and malicious domain names are collected from public sources, and domain WHOIS registration information and Domain Name System (DNS) resolution records are obtained. Secondly, a feature fusion module based on attention is used to fuse the text, registration, and resolution features of domain names. Finally, a malicious domain name detection model called Iconformer is designed, which combines the advantages of attention mechanisms for global modeling and CNN for local modeling. Iconformer incorporates a macaron-style feed-forward module to fully capture multi-level features of domain names. Comparative, ablation, and generalization experiments are conducted on the benign domain name dataset Cisco Umbrella and nine malicious domain name datasets, including CICBellDNS2021, among others. The results demonstrate that this method significantly improves the performance and generalization ability of malicious domain name detection. Compared with the comparative models, the accuracy and precision are improved by 1.17% and 0.94% respectively.

Keywords: Malicious Domain Name Detection; Multi-type features; Feature Fusion; WHOIS Registration Information; DNS Resolution Records

附图索引

附表索引

第1章 绪论

1.1 研究背景与意义

互联网的迅速普及改变了人们的生活、工作和交流方式，然而，隐藏在网络中的恶意事件频发，域名的开放性使其成为网络犯罪的温床，网络犯罪分子利用恶意域名进行恶意活动，例如垃圾邮件、网络钓鱼攻击、分发恶意软件和为僵尸网络托管命令和控制(Command and Control, C&C)服务器等。根据国家互联网应急中心(National Internet Emergency Center, CNCERT/CC)发布的2024年第11期《网络安全信息与动态周报》[1]指出，该周中国境内近112万个IP地址对应的主机被木马或僵尸程序控制，较上月增长22.3%。新增信息安全漏洞数量为2532个，较上月增长19.3%。事件受理方面，CNCERT接收到网络安全事件报告417件，数量最多的分别是恶意软件、垃圾邮件和钓鱼网站。

恶意软件、垃圾邮件和钓鱼网站等通常利用恶意域名作为攻击载体，旨在窃取用户信息、损害系统或网络功能，甚至完全控制被感染的计算机。恶意软件通常采用邮件附件、不法下载等途径传播。而垃圾邮件通常给大量用户发送，包含广告、欺诈信息或恶意链接。钓鱼网站通常伪装成合法的银行、品牌供应商或在线商城，诱骗用户输入账号、密码、信用卡信息等敏感数据。因此，及时检测和阻止恶意域名对于保障互联网的正常运行和保护用户免受网络攻击至关重要。

现有恶意域名检测方法主要分为三类，分别为基于黑名单的方法，基于机器学习的方法和基于深度学习的方法。其中，基于黑名单的检测方法侧重于识别黑名单中的恶意域名，无法有效识别新出现或不在黑名单中的恶意域名。基于机器学习的方法，通常人工提取域名的文本特征和统计特征，例如字母频率、元音比例、数字比例和字符的信息熵等，辅之以流量信息进行检测[2]。该方法与基于黑名单机制的方法相比，效果有一定的提高，但检测效果过于依赖特征工程，部分特征提取较难，需花费大量的时间成本，导致效率较低。随着深度学习在自然语言处理(Natural language Processing, NLP)领域的快速发展，一些学者利用深度学习在文本分类中的优异性能，将深度学习方法应用于恶意域名检测。相较于机器学习方法，深度学习方法不依赖人工提取特征，效率较高，且域名检测效果较好，但现有大多深度学习方法依赖于单一类型特征，如捕获域名字符串的语义特征或分析网络流量的特征等。然而，单一类型的特征无法充分表达恶意域名语义信息，导致针对部分构词复杂的恶意域名检测效果和泛化性较差，因此，如何丰富并融合不同类型的域名特征仍是当前研究的难点与热点。

综上所述，本文旨在探索如何有效地丰富并捕获不同类型域名特征，提升恶意域名检测的效果和泛化性，从而更好地保护网络安全和用户隐私。

1.2 国内外研究现状

域名组成结构表明，合法域名通常由人类容易记忆和理解的单词按照一定规则组合而成，相反，恶意域名通常使用语义不相关的词汇随机组合，可读性较差，如“pansksakrswap.com”[3]。针对此，研究者从多角度分析域名结构以识别恶意域名。目前，国内外针对恶意域名的检测研究主要分为以下三种方法：一是基于黑名单的恶意域名检测；二是基于机器学习的恶意域名检测；三是基于深度学习的恶意域名检测。

1.2.1 基于黑名单的恶意域名检测方法

早期，部分研究者采用黑名单的方法检测恶意域名，这类方法构建并维护一个已知的恶意域名的黑名单，当查询的域名存在于黑名单中时，判断其为恶意域名。Antonakakis 等人[4]设计了一个僵尸网络识别系统 Pleiades，人工构建包含已知恶意域名的黑名单，并实时跟踪域名解析结果以检测僵尸网络。Ma 等人[5]爬取公开的恶意域名信息以自动更新黑名单。随着网络技术的发展，许多专业安全机构公开发布所收集到的大型域名黑名单，供研究者和企业免费使用，如 360 网络安全实验室[1]、FKIE[2]和思科[3]等。基于黑名单的方法实现简单，实时检测效率高，但存在无法有效检测新出现的域名的局限。

1.2.2 基于机器学习的恶意域名检测方法

基于机器学习的方法通常会人工提取域名的字符、结构特征等构建特征集，然后输入到分类模型如支持向量机(Support Vector Machine, SVM)、随机森林等中进行判别，部分研究者采用集成学习算法，将分类模型集成以提高检测效率。Ispahany 等人[6]用支持向量机(Support Vector Machine, SVM)，K 最邻近(K Nearest Neighbor, KNN)，朴素贝叶斯(Naive Bayes)等算法作为分类器，仅使用五个特征，得到 99.2%的准确率，实验结果得出熵值对于部分机器学习模型影响较小。Atrees 等人[7]采用 CfsSubsetEval 算法过滤数据集中的冗余和无关特征，以获得与标签高度相关的特征子集，然后，提出一种 AdaBoost(Adaptive Boosting)集成学习算法，将 J48、Naive Bayes、SVM 和决策树算法与 AdaBoost 算法组合进行实验比较。结果表明，AdaBoost 能够有效提升恶意域名

[1] https://data.netlab.360.com/dga/

[2] https://dgarchive.caad.fkie.fraunhofer.de/

[3] https://s3-us-west-1.amazonaws.com/umbrella-static/index.html

检测效果。Chiong R 等人[8]提出了恶意域名检测模型集成模糊加权最小二乘支持向量机(Ensemble Fuzzy-Weighted Least Squares Support Vector Machine, EFW-LS-SVM)，首先，采用合成少数过采样技术(Synthetic minority oversampling Technique, SMOTE)平衡合法域名和恶意域名数量；然后，引入模糊加权操作，将误差约束赋值以确定每个样本的重要性，以提高最小二乘支持向量机(Least Squares Support Vector Machine, LS-SVM)鲁棒性；最后，使用集成学习方法将多个 LS-SVM 与加权操作整合在一起，缓解过拟合的问题，以进一步提高 LS-SVM 的效果。

集成学习算法需要较大的计算成本，且训练样本需提前标记，故部分研究者提出无监督学习的方法以检测恶意域名。Wang H 等人[9]设计一种了三层动态恶意域名检测方法，提出了域名的三个高维特征：域名形成概率、域名字符出现概率的标准差、前后域名间 js 散度以检测恶意域名。该方法只需要将少量合法域名的这三个特征与一些域名系统(Domain Name System, DNS)特征相结合，不需要对恶意样本进行标记和训练。

域名在解析过程中会产生大量注册和解析记录，部分研究者人工提取注册信息和解析记录特征以构建特征集。马栋林等人[10]将域名字符特征、访问特征和解析特征等 6 种特征整合，并改进 Relief 算法，采用 C5.0 分类器实现检测。Pritom 等人[11]采用随机森林算法根据域名词汇特征、WHOIS 注册信息、域名熵值等 11 种特征对新冠疫情相关恶意域名进行分类，指出域名的 WHOIS 部分注册信息对于检测新冠疫情相关恶意域名具有重要作用。臧小东等人[12]提取 DNS 流量中的生存时间(Time To Live, TTL)值、WHOIS 注册信息、域名的活动历史和 IP 地址的归属分布等作为特征，采用 SVM 作为分类器以检测恶意域名。于光喜等人[13]采用分类分析和聚类分析两阶段对恶意域名进行检测，在分类分析阶段，人工提取域名字符特征并采用随机森林算法进行检测；在聚类分析阶段，采用 X-means 算法将域名字符特征和访问行为特征相结合，有效处理大规模 DNS 日志数据，并从中检测出域名生成算法(Domain Generation Algorithm, DGA)域名。

基于机器学习的恶意域名检测方法效果较好，但是检测效果过于依赖特征工程，存在时间开销大、部分特征对检测结果影响小的问题。

1.2.3 基于深度学习的恶意域名检测方法

随着深度学习的兴起，部分研究者采用深度神经网络进行恶意域名检测，相较于机器学习，深度学习不依赖特征工程，模型学习大量数据样本的潜在特征以实现检测，能够有效减少时间开销。

Woodbridge 等人[14]首次将深度学习技术应用到恶意域名检测任务，采用长

短期记忆网络(Long Short-Term Memory，LSTM)自动提取域名字符串潜在特征，效果优于人工提取域名特征的方法。但文献[14]的数据集中合法和恶意域名的数量差别较大，故 Tran 等人[15]将敏感学习引入 LSTM 的反向传播过程，以缓解恶意域名占比较少时模型效果较差的问题。Zhang 等人[16]首次将卷积神经网络(Convolutional Neural Network, CNN)应用于字符级恶意域名检测任务，提出 NYU 模型。随后 Saxe 等人[17]提出 Inincea 模型，相较于 Inincea，NYU 中 CNN 是堆叠结构，且池化窗口大小是根据域名长度决定的。Chen 等人[18]采用 GoogleNetV1 实现恶意域名检测，实验结果得出，购物网站大多被误判为恶意网站，而办公网站等误判率很低。LE 等人[19]提出一种基于域名字符串中字符级和词级特征的恶意域名检测网络 URLNet。设计了字符级 CNN 和单词级 CNN，以学习域名字符串中字符间和单词间的语义表达。罗赟骞等人[20]结合 LSTM 和 CNN，分别从时间和空间的维度提取域名的潜在特征。Vinayakumar 等人[21]提出 Deep Bot Detection(DBD)模型，采用 CNN 和 LSTM 提取域名的无效域名(Non-Existent Domain, NXDomain)响应、DNS 统计特征和基于时间的 DNS 流量分析等特征进行检测。Peng 等人[22]采用 CNN 和 LSTM 融合域名文本和主机特征等，以检测恶意域名，获得了较好的效果。

2017 年谷歌提出完全基于注意力机制的网络架构 Transformer[23]，替代了传统的 CNN 和循环神经网络(Recurrent Neural Network, RNN)，并在多个任务上取得了最好的效果。之后，大多研究者开始研究将 Transformer 及其变体应用在恶意域名检测任务。Maneriker 等人[24]采用基于 Transformer 架构的 BERT(Bidirectional Encoder Representations from Transformers)和 RoBERTa 预训练模型，然后在恶意域名数据集上微调(fine-tuning)模型参数，实验表明，采用预训练模型可以显著提升恶意域名检测效果，此外，他们研究了恶意域名检测模型的对抗鲁棒性。Transformer 架构中多头自注意力(Multi Head Self-Attention, MHSA)能够有效提取域名全局依赖关系，但是缺乏字符间局部特征。因此，Zhao 等人[25]将 Transformer 与 CNN 结合，检测效果较好，尤其针对单词拼接类恶意域名效果很好。

词嵌入作为 NLP 任务的基础工具，在恶意域名检测中发挥了重要作用。一些研究者采用不同类型的词嵌入方法以提升恶意域名检测效果。Yang 等人[26]提出一种基于 N-gram 和 Transformer 的恶意域名检测模型 N-Trans，在域名字符串开头和结尾处添加标志位，并采用 N-gram 算法处理数据集后，利用 Transformer 模型提取域名特征。Liu 等人[27]采用 CharBERT 预训练模型，以获取域名词嵌入矩阵，并采用金字塔结构分层提取域名特征，然后，层感知注意力模块自主学习不同层次的特征之间的关联，并给每种特征分配权重系数。最后，空间金字塔池化模块对加权多级特征进行多尺度下采样，实现局部特征的捕获以

及全局特征的聚合。Zhang 等人[28]采用词级嵌入方法，并结合 CNN 和 Transformer 以检测恶意域名。张震等人[29]提出一种基于图对比学习的恶意域名检测方法，以域名和 IP 地址作为异构图的两类节点并根据其属性建立对应节点的特征矩阵,依据域名之间的包含关系、相似度度量以及域名和 IP 地址之间对应关系构建 3 种元路径。吴涛等人[30]提出一种迁移自反馈学习的小样本恶意域名检测方法，首先，采用 CNN 和双向长短时记忆神经网络(Bi-directional Long Short Term Memory, BiLSTM)提取域名字符特征和上下文语义信息；然后，将学习到的网络模型参数迁移至小样本的恶意域名检测模型中。余子丞等人[31]采用 Transformer 编码器捕获域名字符的全局信息，通过 CNN 获取不同粒度的长距离上下文特征，同时引入 BiLSTM 和 CNN 得到浅层时空特征，融合长距离上下文特征和浅层时空特征进行 DGA 域名检测。Yang 等人[32]提出一个名为 Fast3DS 的实时恶意域名检测系统。采用并行深度卷积结构以替代标准卷积层，并提出了一个轻量级的全局平均池化以替代全连接层，有效减少模型参数和计算时间开销。为了弥补模型轻量化所导致的精度降低，引入一种轻量级的注意力机制来提高模型检测的准确性。

相较于基于黑名单和基于机器学习的方法，基于深度学习的恶意域名检测方法不需要人工提取特征，效果较好，但是现有大多深度学习方法依赖于单一类型特征，对域名特征的多样性考虑不足，导致对于部分构词复杂的恶意域名检测效果较差。此外，恶意域名生成策略不断变化，现有大多方法对模型的泛化能力的重视程度不足，导致针对新出现的恶意域名检测效果较差。

1.3 主要研究内容和组织结构

1.3.1 主要研究内容

本文针对现有恶意域名检测方法存在的一些挑战，主要包括：挖掘域名多样性特征、有效捕获域名潜在特征和提高模型泛化性。具体研究内容如下：

（1）挖掘域名多样性特征

鉴于现有方法对域名特征的多样性考虑不足，第三章为丰富域名的基本特性，采用 python-whois 工具获取域名 WHOIS 注册信息，构建域名注册特征，融合域名文本特征实现恶意域名检测。第四章为进一步丰富域名的行为模式，采用 dnspython 工具获取域名 DNS 解析记录，从而构建域名解析特征，并融合域名文本和注册特征实现恶意域名检测。

（2）捕获域名潜在特征

鉴于现有方法处理部分构词复杂的恶意域名时效果不佳，第三章采用 Transformer 以提取域名文本的全局特征，并捕获域名文本与注册信息之间的语

义关联，同时，结合 CNN 提取域名局部特征，旨在充分捕获域名潜在特征，以提高恶意域名的检测性能。第四章中，进一步提出 Iconformer 模型，结合了注意力机制的全局建模和 CNN 的局部建模优势，此外，引入马卡龙式前馈模块，用于多级特征提取，从而使模型能够更有效地学习域名序列的潜在特征表示。

（3）提高模型泛化性

鉴于现有方法泛化性不足，第三章在数据方面，着重关注部分重要的 WHOIS 注册信息，忽略部分影响较小信息，从而降低模型的时间开销以提升泛化能力。第四章进一步在数据方面引入域名解析特征，并在模型方面采用多头概率稀疏自注意力(Multi Head ProbSparse Self-Attention, MHPSSA)模块，有效降低模型计算复杂度和参数量。此外，卷积模块结合蒸馏操作，减少冗余信息。通过将两者结合，提高模型泛化能力。

1.3.2 组织结构

本文分为五章，组织结构如图 1.1 所示。

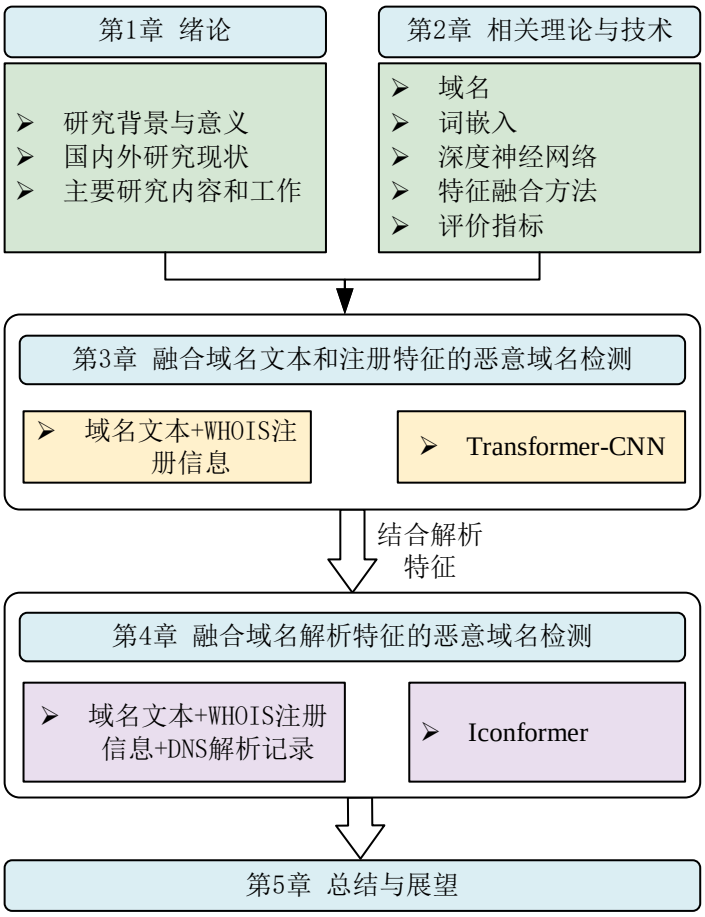


图 1.1 本文组织结构

本文各章概要如下：

第 1 章：绪论。介绍了恶意域名检测任务的研究背景与意义，探讨国内外研究现状，最后说明本文主要研究内容和组织结构。

第 2 章：相关理论与技术。首先，介绍域名相关知识，包括 DNS、域名结构、DNS 解析记录、WHOIS 注册信息等；然后，探讨了不同的词嵌入方法的优缺点；其次，阐述了恶意域名检测中常见的神经网络，包括 CNN、RNN 和 Transformer；最后介绍了评价指标。

第 3 章：融合域名文本和注册特征的恶意域名检测。鉴于当前恶意域名检测方法存在的问题，主要集中在对域名特征的多样性考虑不足、域名潜在特征提取不充分以及泛化性不足。提出一种融合域名文本和注册特征的恶意域名检测方法。首先，从公开渠道收集包括合法和恶意域名，并获取域名 WHOIS 注册信息；其次，结合 Transformer 和 CNN 模型提取域名文本和注册信息的全局和局部特征；最后，融合两种特征，从而实现域名的分类。

第 4 章：融合域名解析特征的恶意域名检测。在第三章的基础上，融合域名文本、注册和解析特征，以实现恶意域名分类。首先，从公开渠道收集合法和恶意域名，并获取域名 WHOIS 注册信息及 DNS 解析记录；其次，采用基于注意力机制的特征融合模块融合域名文本、注册、解析特征；最后，设计了一种恶意域名检测模型 Iconformer，结合注意力机制的全局建模和 CNN 的局部建模优点，并引入马卡龙式前馈模块提取多级特征，使模型更有效地学习域名序列的潜在特征表示。

第 5 章：总结与展望，首先，梳理总结本文的研究内容和成果；其次，指出目前研究存在的不足；最后，展望未来可行的研究工作以及研究方向。

第2章 相关理论与技术

2.1 域名

2.1.1 DNS

DNS 是互联网基础架构的重要组成部分，用于实现域名与其对应的 IP 地址之间的映射关系[33]。DNS 以一种分布式的树状层次结构维护全球范围的域名和 IP 地址映射数据库，将全球各地的 DNS 服务器组织成一个统一的层次解析系统，这些服务器存储了域名与 IP 地址之间的映射关系，并提供域名解析服务。

域名是 DNS 的核心对象，它们构成了网络资源的命名体系[34]。域名是一种由点分隔的字符序列，通常由英文字母(不区分大小写)、数字和分隔符等组成。域名可以用来标识计算机网络中的各种资源，如网站、邮件服务器、FTP 服务器等。域名的结构通常包括顶级域名(Top-Level Domain, TLD)、二级域名(Second-Level Domain)和子域名(Subdomain)，其中 TLD 代表域名的类型或分类，而二级域名则表示特定的网络实体或组织。如域名“www.google.com”，“.com”是顶级域名，“google”是二级域名。

统一资源定位符(Uniform Resource Locator, URL)是互联网上资源的地址标识符[35]。URL 由若干部分组成，包括协议标识符、主机名、资源路径等。其中，协议标识符指定访问资源所采用的协议，如 HTTP、HTTPS、FTP 等；主机名则指定了资源所在的主机或服务器的域名或 IP 地址；路径部分则指定了资源在服务器上的具体位置。例如 URL(https://blog.csdn.net/weixin_53436351)的构成如图 2.1 所示。URL 旨在统一资源的定位方式，使用户可以通过简单的方式访问互联网上的各种资源。

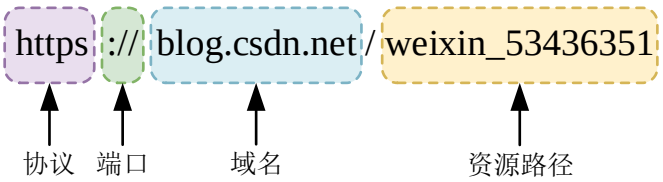


图 2.1 URL 结构

DNS、域名和 URL 之间存在密切的关系。DNS 通过域名解析服务将域名转换为 IP 地址，使用户可以通过简单易记的域名访问互联网上的资源，而无需记忆复杂的 IP 地址。域名作为网络资源的标识符，可以通过 DNS 系统进行解析，从而定位到相应的服务器和资源。而 URL 作为资源的地址标识符，则承载了用户访问资源的具体路径和协议信息，通过 URL，用户可以准确定位到所需资源

的位置并进行访问。因此，DNS、域名和 URL 共同构成了互联网上资源的命名和定位系统，为用户提供了便捷的访问方式，促进了互联网的发展和应用。

2.1.2 DNS 解析记录

当用户在浏览器输入域名 www.google.com 时，DNS 解析过程如图 2.2 所示。

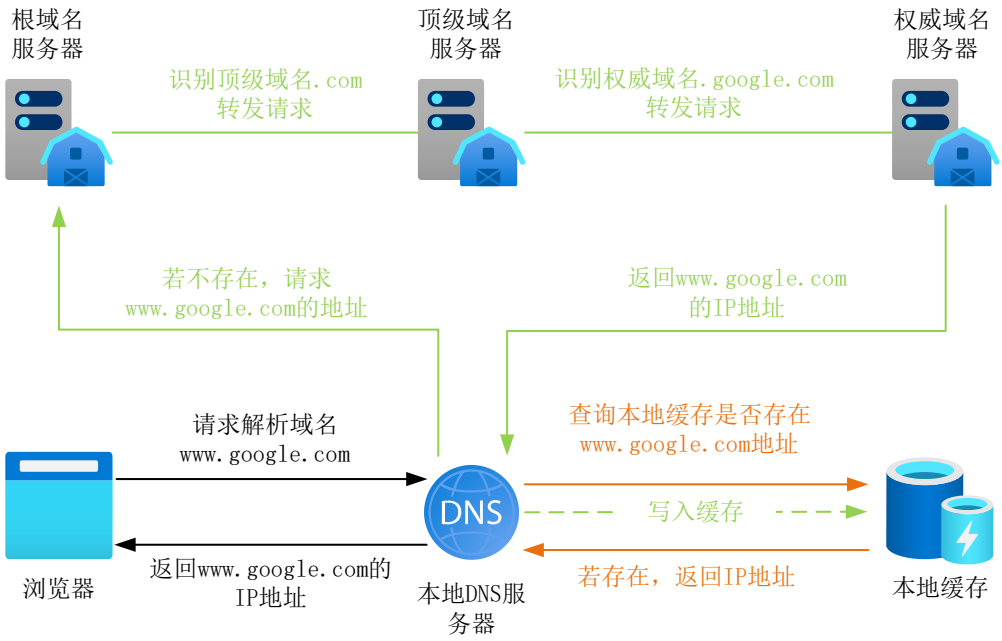


图 2.2 DNS 解析过程

具体流程如下：

（1）本地 DNS 缓存：首先，浏览器通过本地 DNS 服务器检查本地缓存，查看是否已经有该域名对应的 IP 地址的记录。本地缓存存放了最近已请求过的域名和 IP 地址映射，可以加速域名解析过程。

（2）递归查询：如果本地缓存没有找到对应的 IP 地址，用户的设备就会向预配置的递归 DNS 服务器发送域名解析查询请求。递归 DNS 服务器通常由本地互联网服务提供商(Internet Service Provider, ISP)提供。

（3）根域名服务器查询：递归 DNS 服务器发现自己也没有该域名的解析记录，于是向根域名服务器发出查询请求。根域名服务器负责转发查询请求到顶级域(com、net、org 等)的域名服务器。

（4）顶级域名服务器查询：根域名服务器根据域名结构，识别出顶级域，并将查询请求转发给负责该顶级域的域名服务器。www.google.com 会被转发到负责.com 域的域名服务器。

（5）权威域名服务器查询：顶级域名服务器再根据域名的结构，识别出权威域名服务器，并将查询请求转发给该名称服务器。权威域名服务器保存了该域名的实际 IP 地址。

（6）返回查询结果：权威域名服务器将找到的 IP 地址，通过之前的查询路径返回给递归 DNS 服务器。递归 DNS 服务器再返回 IP 地址给浏览器，并缓存结果。

DNS 采用请求-响应模型，其数据交换通过规范化的 DNS 报文进行。DNS 报文包含丰富的功能字段，以支持各种 DNS 操作。根据报文的方向和内容，DNS 报文可分为查询报文、响应报文、通知报文和区传送报文[36]。其中，查询报文由本地 DNS 解析器构造，用于向名称服务器提出域名解析服务的请求，包括要查询的域名、查询类型等。响应报文由名称服务器生成，用于回复解析查询请求，其答复部分包含所查询域名对应的 IP 地址等。通知报文由名称服务器主动下发，以通知本地 DNS 解析器其域名记录的变更。传送报文用于在主从服务器间传递区域数据。相较于查询报文，通知报文和传送报文由名称服务器构造。

响应报文在整个 DNS 解析过程中具有关键作用，因为它包含了名称服务器解析的最终结果，决定 DNS 解析的成功与否。响应报文中，最关键的是回答部分的资源记录(Resource Records, RRs)，包括 RRs 中 RDATA 子字段的 IP 地址、RRs 的 TTL 和类型(RdType)字段。其中，IP 地址提供域名最终映射的网络身份标识，是 DNS 查询结果的核心；TTL 值反映了本地 DNS 服务器对域名记录的缓存时间，影响解析流量和效率。通常，合法域名为提高其网站访问速度，TTL 值一般较大，而恶意域名因其需要经常改变其解析记录的特点，TTL 值一般较小；RdType 说明返回的命名记录类别。这几个字段构成了响应报文的核心，正确解析它们是保证客户端获得预期域名解析结果的关键。

2.1.3 WHOIS 注册信息

域名 WHOIS 注册信息涵盖了域名持有者身份信息以及域名注册情况等关键信息，包含域名的注册人联系信息、域名注册商、注册和到期时间以及域名状态等。因此，获取域名 WHOIS 注册信息对于验证域名真实性和持有者身份至关重要。

在学术研究领域，WHOIS 注册信息被广泛运用于识别域名使用者身份、研究域名注册模式以及探测网络犯罪活动等[10–12,37,38]。通过分析 WHOIS 注册信息中的注册邮箱、注册名称等信息，可以推断出域名持有者的地理分布和机构属性。统计分析大量 WHOIS 注册信息后，有助于揭示一个国家或地区的域名注册概况和变化趋势。

域名 WHOIS 注册信息主要包含以下信息：

（1）域名注册机构信息，包括注册服务机构的名称、联系信息等，可用于确定域名的注册商。

（2）域名注册人信息，包括注册人的姓名、地址、电话和邮箱等联系信息，

可用于确定域名持有者身份。

（3）技术联系人信息，涵盖网站技术管理者的联系方式。

（4）域名状态信息，包括域名的创建时间、更新时间、到期时间以及是否处于锁定状态等基本信息。

（5）域名服务器信息，即使用的域名服务器地址，通常与注册商相关联。

（6）与注册商的关联性信息，一些注册商会在 WHOIS 注册信息中添加自己的标识信息。

2.2 词嵌入

词嵌入(Word Embedding)是 NLP 领域中一种重要的技术手段，用于获取词语的语义信息。其目的在于将词语映射到一个稠密的连续向量空间中，使语义相关的词语在空间中的距离较近[39]。词嵌入的核心在于理解语境信息，赋予词语语义上的表示，即将相似语境中出现的词语映射到相近的向量空间位置。目前，常见的词嵌入方法包括独热编码[40]、Word2Vec[41]以及 BERT 预训练模型[42]等。词嵌入向量包含词语的语义信息，可广泛应用于 NLP 的下游任务上，如文本分类、情感分析、语义解析等。

2.2.1 独热编码

独热(One-hot)编码是一种简单而有效的特征编码方式，用于将类别特征转换为向量表示。该编码方式为每个类别创建一个独立的二值变量，其中该类别对应的变量取值为 1，而其他类别的变量取值为 0。这种编码方式实现了对不同类别之间的正交编码表达。

One-hot 编码的优势在于，首先，它可以避免模型将类别值的大小关系作为重要信息；其次，One-hot 编码将类别特征转化为数值型特征，使得模型能够处理；此外，编码后的特征之间不存在相关性，有助于减少特征冗余。因此，One-hot 编码被广泛应用于 NLP 任务中。

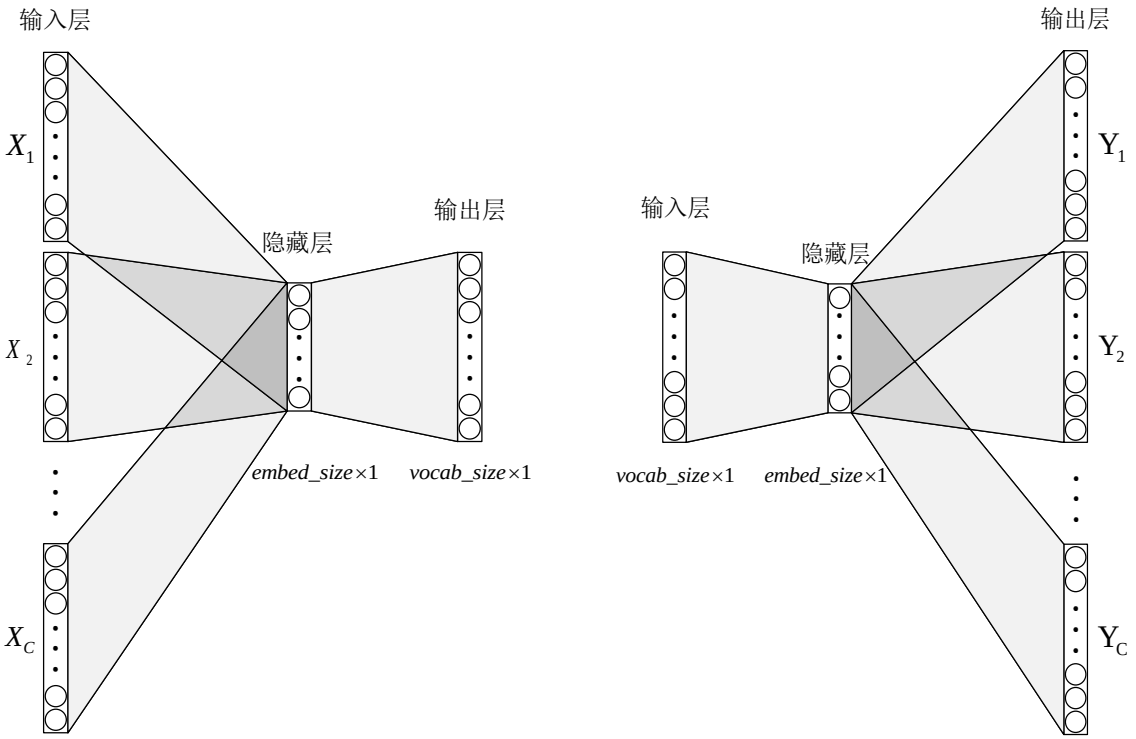
然而，One-hot 编码也存在一些潜在的缺点。首先，它导致特征维度显著增加，增加存储和计算成本；其次，编码后的特征是独立存在的，丢失了类别之间的潜在相关性信息；此外，当新的类别加入时，需要重新定义编码字典，限制了编码的灵活性。

2.2.2 分布式词嵌入

（1）Word2Vec

Word2Vec 由 Tomas 等人在 2013 年提出，旨在通过神经网络模型学习得到词语的连续稠密向量表示，使得语义相关的词向量距离较近。Word2Vec 包含两种模型，连续词袋模型(Continuous Bag-of-Words, CBOW)和 Skip-gram 模型，

CBOW 和 Skip-gram 的模型结构如图 2.3 所示。



(a) CBOW　　(b) Skip-gram

图 2.3 CBOW 和 Skip-gram 模型结构

如图 2.3 中(a)所示，CBOW 模型是根据上下文单词去预测目标词以训练得到词向量，过程如下：

1）输入层：接收 One-hot 张量 $V \in \mathbb{R}^{1\times vocab\_size}$ 作为网络的输入，其中 $vocab\_size$ 为字典大小，V 包含当前句子中上下文单词的 One-hot 表示。

2）隐藏层：将张量 V 乘以一个词嵌入张量 $W^1 \in \mathbb{R}^{vocab\_size\times embed\_size}$ 其中 $embed\_size$ 为词嵌入维度，得到一个形状为 $\mathbb{R}^{1\times embed\_size}$ 的张量，作为隐藏层的输出。

3）输出层：将隐藏层的结果乘以另一个词嵌入张量 $W^2 \in \mathbb{R}^{embed\_size\times vocab\_size}$ 得到形状为 $\mathbb{R}^{1\times vocab\_size}$ 的张量。这个张量经过 softmax 变换后，就得到了当前上下文对中心的预测结果。

图 2.3 中(b)为 Skip-gram 模型，是根据目标词去预测上下文以训练得到词向量，过程与 CBOW 模型相反，CBOW 模型和 Skip-gram 模型都使用随机梯度下降等优化算法更新模型参数，以学习到单词的词向量。相较于传统词袋模型，Word2Vec 学习的词向量包含语义信息，可应用于 NLP 中句法分析和文本分类等任务中，目前 Word2Vec 已成为获取分布式词嵌入表示的重要方法之一。

（2）BERT

BERT[42]是由谷歌公司于 2018 年提出的一种词嵌入预训练模型，旨在学习语料库中词语的高质量词嵌入，该模型在机器阅读理解、GLUE 基准测试等多个测试中超越人类的表现。

BERT 基于 Transformer 编码器结构，通过在大规模文本数据上预训练以捕捉语言的深层双向表征，然后再针对不同的 NLP 任务进行微调，如文本分类、问答系统等。与仅考虑单向上下文的词嵌入方法不同，BERT 在预训练过程中同时考虑了词语前后的上下文，使得学习到的词向量不仅编码了词语的语法特征，还包含了丰富的前后文语义信息。

此外，BERT 的预训练任务包括遮蔽语言模型(Masked Language Model, MLM)和句子关系预测(Next Sentence Prediction, NSP)，MLM 任务中，模型被训练来预测输入句子中被遮蔽的词；而在 NSP 任务中，模型需要判断两个句子是否是连续的文本序列。

2.3 深度神经网络

2.3.1 CNN

NLP 领域，CNN[43]在文本处理和语义建模方面取得了显著的成就。设计之初，CNN 旨在处理图像数据，但其优秀的特征提取能力和并行计算的特性使其在 NLP 任务中也得到了广泛地应用。CNN 的核心概念源自图像处理中的滤波器(filter)概念，通过滑动窗口(kernel)的方式提取不同位置的特征，从而有效地捕获文本序列的局部特征。

CNN 被广泛应用于文本分类、情感分析和命名实体识别等任务。文本分类任务中，CNN 通过学习局部特征和全局特征以判别文本的类别，情感分析任务中，CNN 可以从句子中提取情感词和语境信息，以分析文本的情感倾向；而恶意域名检测任务中，CNN 通过学习域名中字符间的关系，以判断域名是否为恶意。

CNN 通常由卷积层和池化层构成。其中卷积层用于捕获局部信息和提取不同长度的特征，而池化层则用于降维和保留显著特征。此外，为了适应不同的任务需求，CNN 还会结合其他组件，如全连接层、丢弃层等。

（1）卷积层

一维 CNN 结构如图 2.4 所示。假设卷积核大小为$h\times d$，h是卷积核尺寸，卷积核定义为$F\in\mathbb{R}^{h\times d}$，输入特征维度为矩阵$V\in\mathbb{R}^{s\times d}$，其中$s\in S$表示字符串序列词典总长度，$d$表示每个字符的向量维度。则一维 CNN 层的计算公式如式(2.1)所示。

$$C_i = f(F\cdot V_{i:i+h-1}+b) \tag{2.1}$$

其中，C_i表示通过卷积后第i个字符的局部特征，f为非线性激活函数，$V_{i:j}$表示字符串中第i个字符到第j个字符的向量矩阵，b是偏置量。

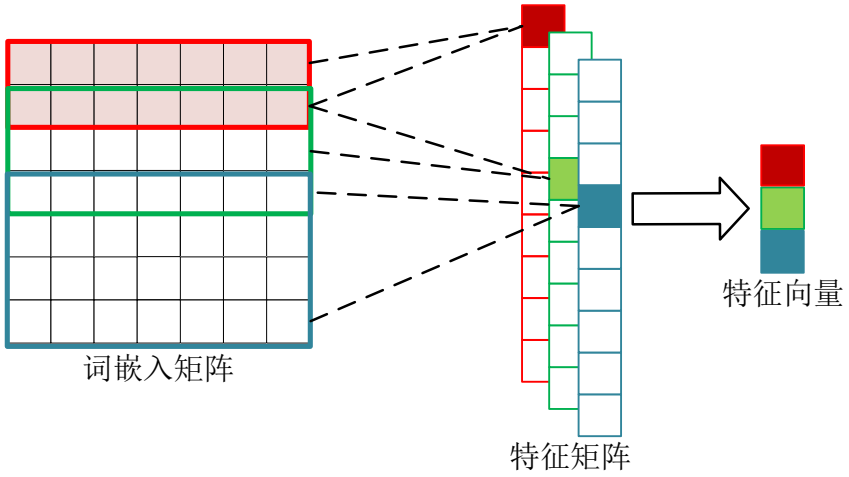
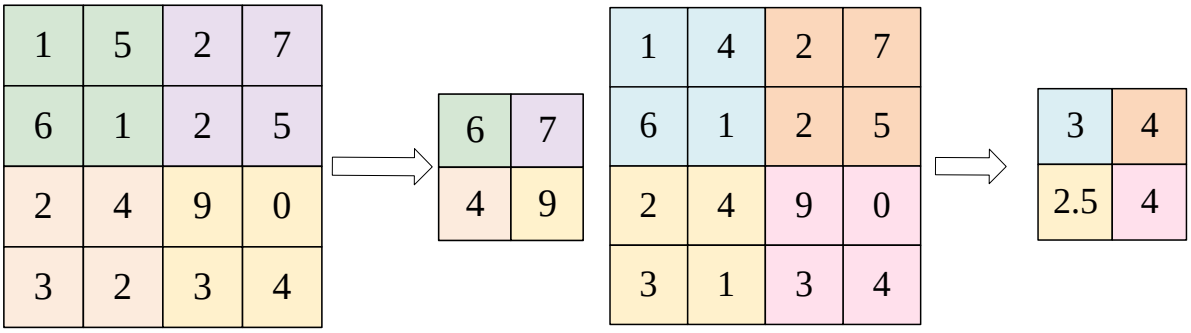


图 2.4 CNN 结构

（2）池化层

池化层(Pooling Layer)通常位于卷积层之后，旨在将卷积层输出的特征图进一步抽象和精炼。常见的池化操作包括最大池化(Max Pooling)和平均池化(Average Pooling)等。最大池化采用一种非线性下采样方法，将特征图划分为多个不重叠的矩形区域，提取每个子区域的最大值。从而捕捉显著特征。而平均池化则计算每个子区域内的平均值，以保留特征的整体信息。平均池化和最大池化过程如图 2.5 所示。



(a) 最大池化　　(b) 平均池化

图 2.5 最大池化和平均池化过程

NLP 任务中，卷积层后的特征图通常反映了不同通道上文本的高级抽象语义特征，池化层有助于提取出这些显著语义特征，同时减少特征图在时序或空间维度上的大小，使得后续的全连接层的输入更加紧凑。此外，池化操作还赋予了模型平移不变性，增强模型对语义顺序变化的适应能力。

2.3.2 RNN

（1）RNN

RNN[44]旨在有效处理时序数据，如文本、音频、视频等，在 RNN 中，信息的传递是通过循环连接实现的，这种连接机制使得网络能够对序列数据进行建模。RNN 通过将当前时刻的输入与上一时刻的隐藏状态结合，生成当前时刻的隐藏

状态，从而实现对序列信息的记忆和学习。递归结构使得 RNN 能够捕捉序列中的依赖关系，并在处理时序数据时表现出色。RNN 结构如图 2.6 所示。

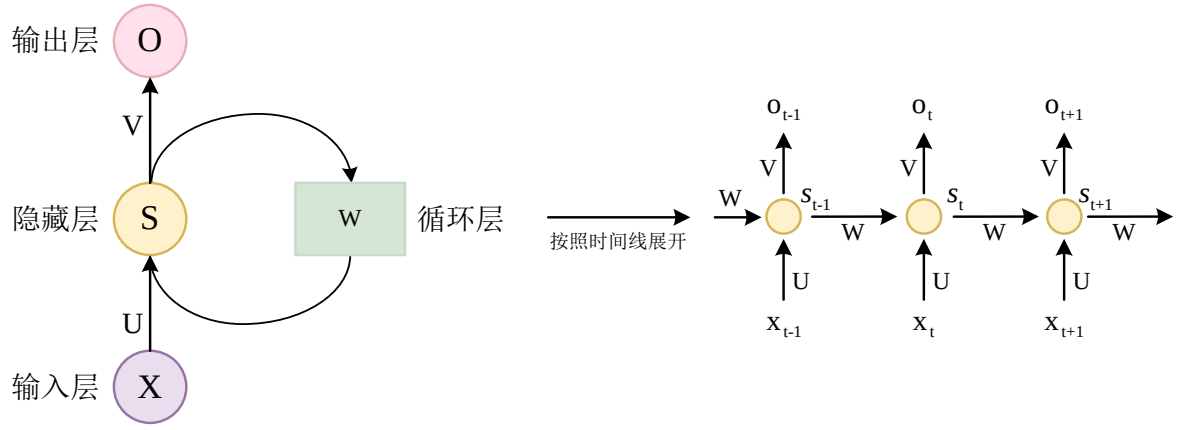


图 2.6 RNN 结构

RNN 包含输入层、隐藏层和输出层。隐藏层中的神经元通过时间步(time step)展开，每个时间步的隐藏状态都依赖于上一个时间步的隐藏状态和当前时间步的输入。这种结构使得网络可以处理不定长度的序列数据，并在训练过程中自动学习序列中的模式和规律。

当 RNN 在t时刻接收到输入之后，隐藏层的值为S_t，输出值为O_t，最为关键的是，S_t的值不仅取决于X_t，还取决于S_{t-1}。输出O_t的计算公式如式(2.2)所示，隐藏层S_t的计算公式如式(2.3)所示。

$$O_t = g\left(V \cdot S_t\right) \tag{2.2}$$

$$S_t = f\left(U \cdot X_t + W \cdot S_{t-1}\right) \tag{2.3}$$

其中，U,V,W为权重矩阵，g,f为非线性激活函数。

然而，RNN 因层与时间步之间计算方式为乘法，从而存在梯度消失和梯度爆炸等问题，限制了其在长序列数据上的表现。为了解决这一问题，LSTM 和门控循环单元(Gated Recurrent Unit, GRU)等变体被提出。这些模型通过引入门控机制，有效地控制了信息的传递和遗忘，从而提高了网络对长序列数据的建模能力。

（2）LSTM

LSTM 网络[45]由 Graves 于 2012 年提出，是一种特殊的循环神经网络结构，其设计灵感源于神经心理学中有关记忆和遗忘的理论，旨在解决 RNN 中存在的梯度消失和梯度爆炸等问题。

与 RNN 不同，LSTM 引入了三个关键的门控单元：输入门(Input Gate)、遗忘门(Forget Gate)和输出门(Output Gate)，以及一个状态单元(Cell State)，用于控制信息的流动和记忆。在 LSTM 中，输入门决定哪些信息将被加入到状态单元

中；遗忘门决定哪些信息将被从状态单元中遗忘；输出门决定哪些信息将被输出到下一层或最终的输出。LSTM 结构如图 2.7 所示。

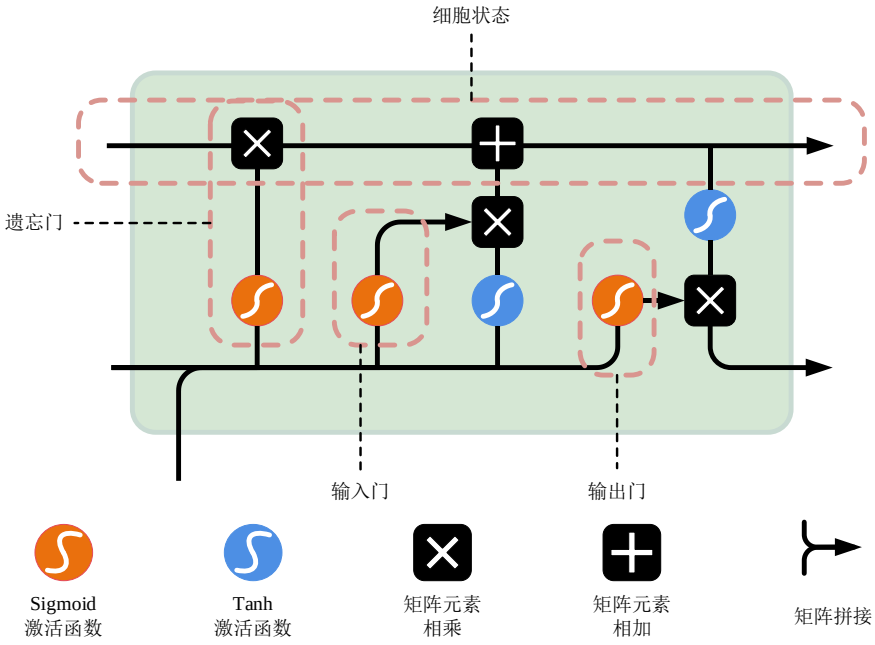


图 2.7 LSTM 结构

1）遗忘门：遗忘门负责控制遗忘上一时刻细胞状态中的信息。它利用前一个隐藏状态与当前输入信息，经过 Sigmoid 函数，输出值范围在 0 到 1 之间。接近 0 表示应该忘记，接近 1 的值表示应该记住。

2）输入门：输入门的任务是更新细胞状态。首先，通过 Sigmoid 函数处理前一时刻隐藏状态和当前输入信息，以确定需要更新的信息；然后，利用 Tanh 函数处理同样的信息，得到一个候选值；最后，将 Sigmoid 输出与 Tanh 输出相乘，决定哪些信息是重要的且应该被保留。

3）细胞状态更新：细胞状态通过遗忘门和输入门进行更新。遗忘门的输出与前一时刻细胞状态相乘，丢弃不需要的信息。然后，加上输入门的输出，将新的重要信息添加到细胞状态中。

4）输出门：输出门负责控制下一时刻隐藏状态的生成。它利用前一时刻隐藏状态与当前输入，经过 Sigmoid 函数确定哪些信息应该传递给下一个时刻。然后，利用 Tanh 函数处理细胞状态，再将两者相乘得到隐藏状态。最终，将隐藏状态作为当前细胞的输出，并将新的细胞状态和隐藏状态传递到下一个时间步。

假设当前时间 t 隐藏单元为 C，输入为 x_t，上一时间步的隐藏状态为 C_{t-1}，时间 t 时刻门定义为：输入门 i_t，遗忘门 f_t，输出门 o_t，具体计算如式(2.4)-(2.6)所示。

$$i_t = \sigma(W_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i) \tag{2.4}$$

$$f_t = \sigma(W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f) \tag{2.5}$$

$$o_t = \sigma(W_o \cdot [C_t, h_{t-1}, x_t] + b_o) \tag{2.6}$$

其中，σ 为非线性激活函数，W_i, W_f, W_o 为权重矩阵，b_i, b_f, b_o 为偏置参数。

LSTM 网络能够捕捉到序列数据中的长期依赖关系，并将这些信息有效地传递到后续的网络层中。因此，在 NLP 任务中，LSTM 被广泛应用于文本生成[46–48]、机器翻译[49–51]、情感分析[52–54]等任务中，取得了显著的效果。

2.3.3 Transformer

（1）Transformer 编码器

Transformer 模型[23]创新的将传统的 RNN 替换为自注意力机制，解决了 RNN 无法并行计算、梯度消失和长距离信息缺失的问题，有效提升了模型效果。Transformer 包含 N 层编码器，每层包含两个子层，即多头自注意力层(Multi-Head Self-Attention, MHSA)和点式前馈网络层(Point wise feed forward network, FFN)。每个子层包含残差连接并通过层归一化，残差连接有助于避免深度网络中的梯度消失问题。Transformer 编码器的结构如图 2.8 所示。

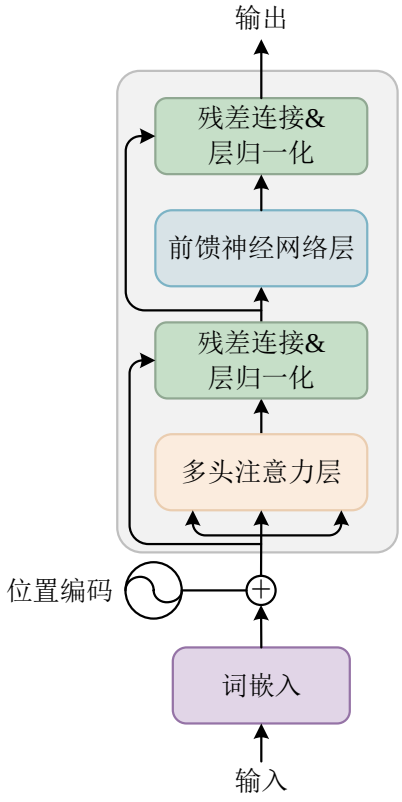


图 2.8 Transformer 编码器结构

（2）位置编码

由于 Transformer 并不包括任何的循环或卷积，所以无法捕获序列顺序信息，

因此嵌入向量进入编码器之前，位置编码向量被加到嵌入向量中，为模型提供单词在句子中相对位置的信息[55]。位置编码通过向输入序列中的每个位置添加特定的向量，使得模型能够在每个位置区分不同的单词。这些位置编码向量利用正弦和余弦函数的周期性特性，以及位置序号和模型隐藏单元数的关系，为输入序列中的单词添加了位置信息，相对位置编码公式如式(2.7)所示。

$$\begin{aligned} PE_{(pos,2i)} &= sin(pos/10000^{2i/d_{model}}) \\ PE_{(pos,2i+1)} &= cos(pos/10000^{2i/d_{model}}) \end{aligned} \tag{2.7}$$

其中，pos表示当前单词所处位置，$pos \in \{0,1,2,\ldots,L\}$，$d_{model}$表示序列词嵌入长度，$i$表示字符所处位置，$i \in \{0,1,2,\cdots,d_{model}\}$。

（3）多头注意力层

多头注意力层包含线性层、按比例缩放的点积注意力和拼接，多头注意力层结构如图 2.9 所示。

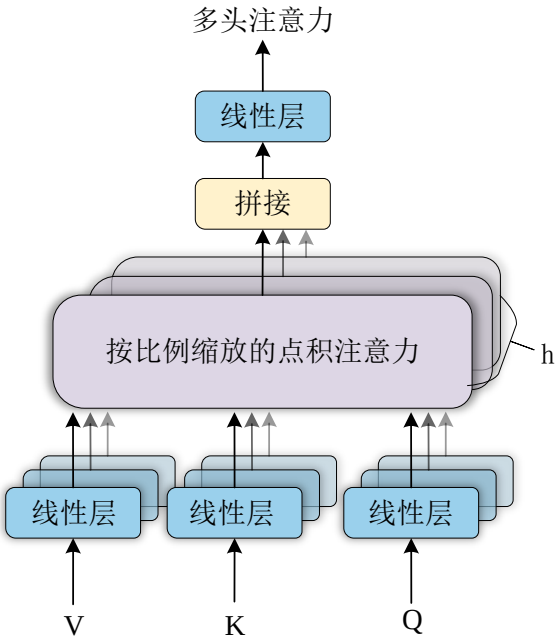


图 2.9 多头注意力层结构

每个多头注意力块有三个输入：Q(请求)、K(主键)、V(数值)，首先经过线性(Dense)层，并拆分成多头，而非单个的注意力头，因为多头允许模型共同注意来自不同表示空间的不同位置的信息。拆分后，每个头部的维度减少，因此，总的计算成本与单个注意力头相同。之后，按比例缩放计算点积以获得注意力分数，计算公式如式(2.8)所示。

$$\text{Attention}(Q,K,V) = \text{softmax}(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}})V \tag{2.8}$$

其中，d_k为向量K的维度，$d_k = d_q$，而除以$\sqrt{d_k}$的原因为防止输入维数过高时，QK^{T}值过大导致 softmax 函数反向传播时发生梯度消失，且不能使QK^{T}值过度增加。

最后经过多头合并及最后一层线性层后输出，多头合并计算公式如式(2.9)所示。

$$\begin{aligned} &\mathrm{Multihead}(Q,K,V) = \mathrm{Contact}(head_1,\ldots,head_h)W^O \\ &where\ head_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.9}$$

其中，$head_i$为第i个$head$的输出，$W_i^Q \in \mathbb{R}^{d_{model}\times d_q}$，$W_i^K \in \mathbb{R}^{d_{model}\times d_k}$，$W_i^V \in \mathbb{R}^{d_{model}\times d_v}$，$W^O \in \mathbb{R}^{hd_v\times d_{model}}$。

（4）前馈神经网络层

点式前馈网络由两层全连接层组成，两层之间有一个 ReLU 激活函数。

（5）残差连接及层归一化

残差连接及层归一化模块连接在多头注意力层和前馈神经网络层后，因此，Transformer 编码器输出计算公式如式(2.10)所示。

$$Outputs = \mathrm{LayerNorm}(inputs + \mathrm{Sublayer}(inputs)) \tag{2.10}$$

其中，$\mathrm{Sublayer}(inputs)$为每个子模块的输出。

2.4 特征融合方法

特征融合作为当前机器学习研究的焦点之一，旨在通过整合多个特征子集来构建新的特征表示，以克服单一特征所带来的表征限制特征融合可在数据预处理阶段进行，亦可在模型层面融合不同特征。常见的特征融合方法包括基于特征组合和基于注意力机制等。相较于单一特征，融合特征能够更丰富地编码样本信息，有效提升模型性能。

2.4.1 特征组合

基于特征组合方式的特征融合方法包含特征拼接和特征求和等，其中，特征拼接和特征求和方式的融合过程如图 2.10 所示。

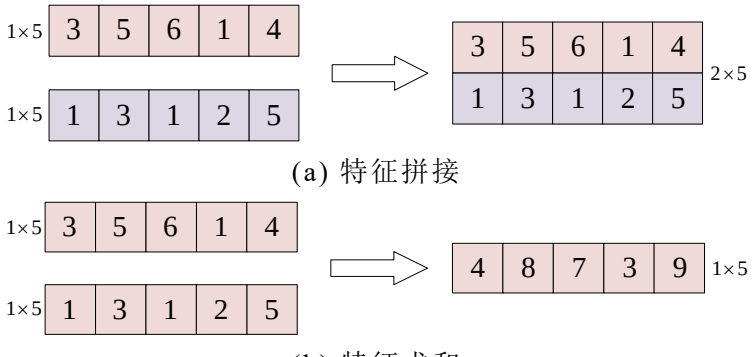


图 2.10 特征拼接和特征求和的融合过程

（1）特征拼接

特征拼接作为特征融合中最简单而常用的方法之一，直接将多个特征向量按一定的维度拼接形成新的特征向量，以充分利用不同特征向量所蕴含的信息。特征拼接操作中，要求原始特征向量具有相同的维度，并需对不同特征进行规范化处理。

特征拼接的优势在于简洁高效，几乎无需调整参数即可实现，易于实施。然而，拼接方法也存在一定的局限性，首先，不同特征的冗余信息或相互冲突的信息会影响特征表达的效果；此外，较高维度的特征，拼接会大幅增加计算复杂度。

（2）特征求和

特征求和将多个特征向量在相应维度上的数值求和以生成新的特征向量。相较于特征拼接，特征求和能够减少特征空间的维度，从而降低数据冗余，并抑制单一特征的噪声，凸显多个特征综合表达的优势。此外，特征求和还保留了原始特征的可加性，有助于模型结果的解释。然而，特征求和存在不同特征的数值范围不一致可能导致信息丢失的问题。因此，特征求和时，必须对各个特征向量进行规范化处理，以确保它们映射到可比较的数值范围内。同时，选择具有较强表达能力且相互补充的特征进行求和，有助于生成更具区分性的融合特征。

2.4.2 注意力机制

近年来，注意力机制[56]在深度神经网络中得到广泛应用，其核心理念在于动态给不同的输入赋予权值，以关注较重要的特征，这一概念也被引入到特征融合领域。基于注意力的特征融合方法旨在动态学习不同特征的权重系数，随后以加权组合的方式进行融合。相较于特征求和或简单拼接，基于注意力机制的特征融合方法更灵活，能够自动关注当前样本更为重要的特征[57]。基于注意力机制的特征融合方法不仅提升了模型对关键特征的感知能力，也有助于抑制无关信息的干扰，基于注意力机制的特征融合方法，使模型能够更加准确地进行预测或分类，为复杂任务的解决提供了有效的手段。

2.5 评价标准

通常，构建混淆矩阵以衡量分类模型性能，恶意域名检测为二分类任务，故混淆矩阵如图 2.11 所示。其中横轴为预测标签，纵轴为真实标签，对角线元素表示每个类的正确分类的数量，而非对角线元素表示不正确分类的数量。

根据混淆矩阵的值，可以计算准确率(Accuracy)、精确率(Precision)、召回率(Recall)及 F1 值(F1-score)等指标，以衡量恶意域名检测模型。其中，准确率表示模型预测正确的样本占全部样本的比例。精确率表示模型预测为正类的样本中真正为正类的比例。召回率表示真实为正类的样本中被正确预测为正类的比例。

F1 值则同时考虑了精确率和召回率，是两者的调和平均数。各项指标计算公式如式(2.11)-(2.14)所示。

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

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

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

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

其中，TP(True Positive)表示模型预测为恶意域名且真实为恶意域名的样本数；FP(False Positive)表示预测为恶意域名但真实为合法域名的样本数；TN(True Negative)表示预测为合法域名且真实为合法域名的样本数；FN(False Negative)表示预测为合法域名但真实为恶意域名的样本数。

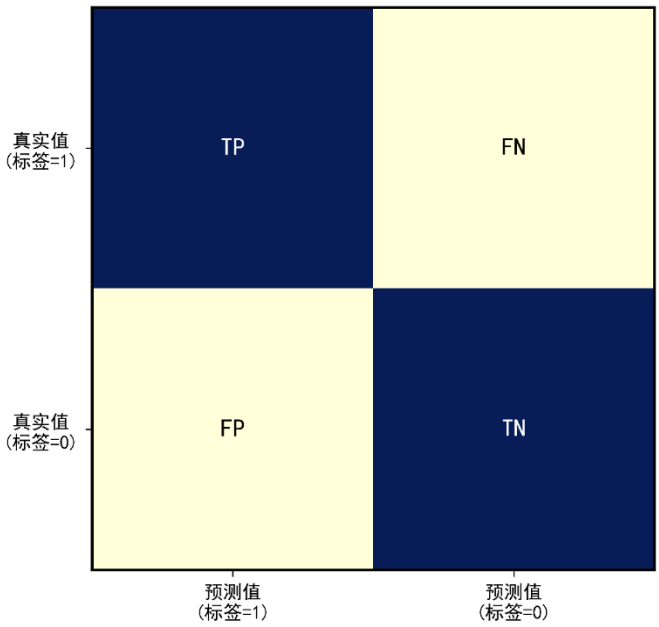


图 2.11 混淆矩阵

2.6 本章小结

本章主要介绍了恶意域名检测相关的基础理论和技术。第一节介绍了域名的相关知识，包含 DNS、DNS 解析记录、WHOIS 注册信息。第二节分析了独热编码和分布式词嵌入方法的优缺点。第三节介绍了恶意域名检测常用的神经网络，包含 CNN、RNN 和 Transformer 网络。第四节阐述了两种不同的特征融合方法，分别为基于特征组合的和基于注意力机制的特征融合。第五节介绍了恶意域名检测模型常用的性能评价指标。

第3章 融合域名文本和注册特征的恶意域名检测

3.1 引言

域名通过 DNS 解析以定位相应的服务器和资源，产生的注册和解析记录信息丰富多样，然而，仅关注域名字符串本身，而不考虑解析过程中的其他信息，将难以全面理解域名的多样性特征。此外，恶意域名的生成通常采用复杂的构词策略，包括随机组合单词或细微的拼写错误，欺骗性较强，如果未能充分捕捉域名潜在的字符组合特征，将难以有效识别构词较复杂的恶意域名。同时，网络犯罪分子的攻击策略不断变化，恶意域名的生成也具有较强的随机性，仅依赖域名构词特性或盲目地加深模型，将导致模型泛化性不足，难以有效识别新出现的恶意域名。

因此，本章提出一种融合域名文本和注册特征的恶意域名检测方法。首先，从公开渠道收集合法和恶意的域名，并获取域名 WHOIS 注册信息，构建域名注册特征，以丰富域名特征表示；然后，忽略部分不关键的 WHOIS 注册信息，增强泛化能力；最后，为了充分捕获域名潜在的特征，采用 Transformer 模型提取域名文本的全局特征并捕获其与注册信息的语义关联，结合 CNN 提取域名文本和注册的局部特征。

3.2 模型结构

融合域名文本和注册特征的恶意域名检测模型结构如图 3.1 所示，包含输入层、特征提取层、特征融合层和输出层。

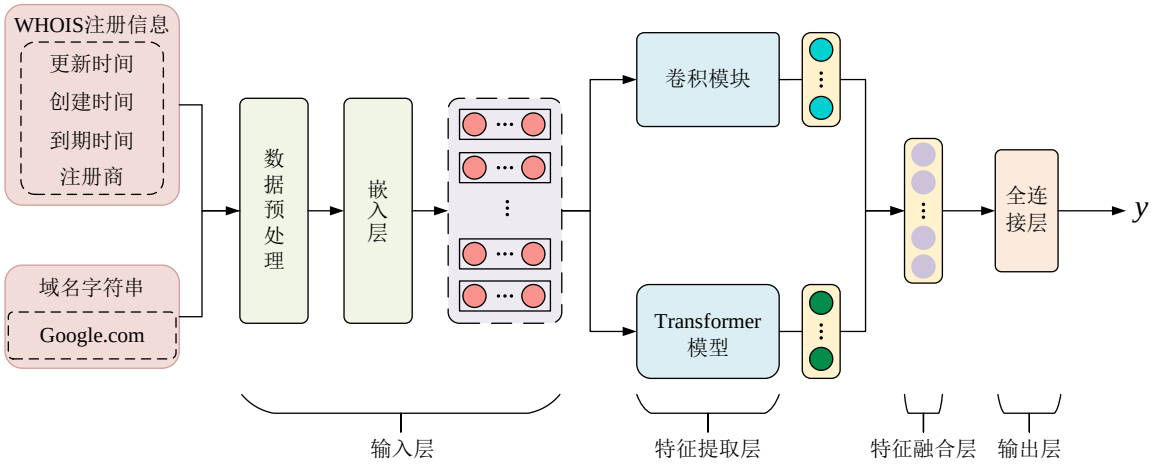


图 3.1 融合域名文本和注册特征的恶意域名检测模型结构

首先，预处理输入域名样本和 WHOIS 注册信息，包含筛除重复域名等；其次，输入层将域名字符串及 WHOIS 注册信息转换为能被深度学习模型处理的数值向量，包含词典构造等；然后，特征提取层采用 Transformer 模型提取域名文本全局特征，以及域名文本与注册信息之间的语义关联，结合卷积模块以提取域名文本和注册信息的局部特征；此外，特征融合层拼接全局特征与局部特征。最后，输出层将域名特征向量经过全连接层后得到域名检测结果 y，实现恶意域名检测。

3.2.1 输入层

首先，将数据集中域名字符串大写字母转换为小写字母，并过滤重复域名；其次，统计域名数据集中出现的字母、数字和特殊字符及 WHOIS 注册信息中注册时间、操作更新时间、到期时间和注册商，以构建长度为 75836 的字典 S，其中域名字符、WHOIS 注册信息作为键，one-hot 向量作为索引值。

然后，统计数据集中最长域名字符长度为 110，WHOIS 注册信息长度为 4，故将融合域名向量长度 L 设置为 114，将长度不足 L 的域名在域名前端填充“0”字符，并根据字典得到长度为 L 的融合域名字符串向量，如式(3.1)所示。

$$Domain=\left[0,0,\ldots,0,a_1,a_2\ldots a_L,w_1,w_2,w_3,w_4\right] \tag{3.1}$$

其中，a_i 为域名字符串中第 i 个字符，w_i 为 WHOIS 中第 i 个注册信息，$a_i,w_i\in S$。

最后，选取 Word2Vec 算法作为词嵌入方法，以生成融合域名词嵌入向量，如式(3.2)所示。

$$Vec\in\mathbb{R}^{L\times d} \tag{3.2}$$

其中，Vec 表示融合域名向量经过嵌入表示后的词向量，$\mathbb{R}$ 表示实数集，L 表示域名长度，d 表示嵌入向量维度。具体 Word2Vec 算法流程见 2.2.2 节。

3.2.2 特征提取层

（1）Transformer

近来，基于注意力机制的 Transformer 模型因其能够出色捕获长距离依赖的特性而在 NLP 领域得到了广泛应用[58,59]。因此，本节采用 Transformer 模型，旨在充分利用其自注意力机制的优势以提取域名文本的全局特征，并捕获域名文本与注册信息之间的语义关联。Transformer 模型具体结构及过程详见 2.3.3 节。

特征提取过程中，Transformer 模型结合自注意力机制和位置编码，使模型可以关注输入序列中不同位置的信息，并根据内容动态地调整注意力权重，从而使模型更关注与当前处理位置相关的信息。因此，Transformer 模型能够有效地捕获文本序列中的长距离依赖关系，从而更好地表征域名文本的全局特征，并捕获域名文本与注册特征之间的语义关联。

此外，相较于传统的 RNN 等，Transformer 模型不需要顺序地传递状态，能够同时处理输入域名序列中的所有词元，各个时间步的计算基本独立。因此，Transformer 模型因其能够并行计算的优势可以高效处理大规模数据，从而降低训练和预测时间开销。

（2）卷积模块

传统的域名文本特征提取方法主要依赖于提取域名字符串的 N-gram 特征，以反映合法域名与恶意域名在不同长度字符组合上的差异[60]。然而，随着 N 值的增加，特征维度呈指数级增长，导致计算开销显著增加。为了解决这一问题，本节采用 CNN，旨在采用尺寸为 2、3 和 4 的卷积核对域名词嵌入向量进行卷积计算，以捕获域名文本和注册信息的局部特征，卷积模块结构如图 3.2 所示。CNN 的具体结构和运作机制详见 2.3.1 节。

在 CNN 中，每个卷积核都能够独立地处理域名文本中的不同部分，从而实现了计算过程的高度并行化。因此，采用 CNN 能够有效降低计算开销，并加速模型的训练和预测过程。此外，为了进一步优化模型性能，采用最大池化层来保留关键特征。最大池化操作能够从局部特征中保留最显著的部分，有效减少特征维度，降低模型的参数量，并有助于防止模型的过拟合现象的发生。

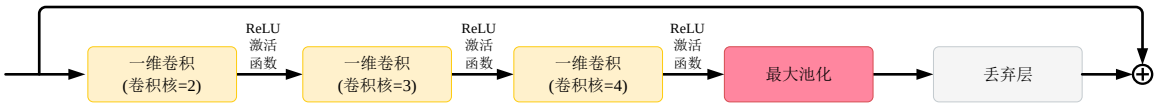


图 3.2 卷积模块结构

3.2.3 特征融合层和输出层

特征融合层在域名文本处理中扮演着关键的角色，旨在有效地整合来自 Transformer 模型和 CNN 的域名特征向量，以提升模型对域名语义和结构的表征能力。本节采用基于特征组合中的拼接方法作为特征融合的策略，具体过程详见 2.4.1 节。

首先从 Transformer 和 CNN 分别获得全局和局部特征向量，然后将它们沿着特征维度进行拼接，形成一个整合的特征向量。这种方法能够有效地保留每个模型提取的特征信息，从而充分利用两种模型的优势，并在一定程度上弥补彼此的不足。

设特征提取层输出的特征向量分别为$V_{Transformer}$和V_{CNN}，则经过特征融合层后得到域名特征向量V_{domain}，计算公式如式(3.3)所示。

$$V_{domain}=\left[V_{Transformer},V_{CNN}\right] \tag{3.3}$$

其中，$[\cdot]$表示拼接操作。

输出层将域名特征向量V_{domain}经过丢弃层，最后采用激活函数为 Sigmoid 的全连接层，输出域名判别结果。

3.3 实验设计与结果分析

3.3.1 数据集

本节构建数据集D，包含合法域名D_{benign}和恶意域名$D_{\text{malicious}}$两类。合法域名D_{benign}通过两种方式收集，首先，人工定义新冠疫情相关关键词，使用谷歌搜索引擎、必应搜索引擎、百度搜索引擎搜索，提取前 200 个网页域名作为合法域名；此外，从 Cisco Umbrella 公开的 100 万合法域名列表[4]中随机抽取。新冠疫情相关搜索关键词列表如表 3.1 所示。

表 3.1 与新冠疫情相关的关键词

COVID-19	COVID	sarscov2	Nucleic Acid
coronavirus	COVID19	Mask	quarantine
corona-virus	corona	sars-cov2	sars

恶意域名$D_{\text{malicious}}$主要来自 Domain Tools 和 ProPrivacy[5]公开的新冠疫情相关域名列表。Domain Tools 列表中每个域名通过 Domain Tools 工具进行预测风险评估后打分，本节选取评分 90 分及以上的域名。经过收集筛选，最终的新冠疫情相关域名数据集包含 3 万条合法域名，2.4 万条恶意域名。

域名的 WHOIS 注册信息包括域名、注册商、续费时间、注册时间、到期时间等。与新冠疫情相关的恶意域名一般在注册后不久就会发起攻击[61]。因此，WHOIS 注册信息对检测新冠疫情相关域名的影响更大。然而，收集完整的注册信息会增加模型的计算量和时间成本。此外，部分注册信息对新冠疫情恶意域名检测的影响较小。因此，采用 python-whois 工具获取数据集中每个域名的 WHOIS 注册信息，相关代码详见附录 A。并只保留与时间相关的信息和注册商信息。所收集 WHOIS 注册信息的描述如表 3.2 所示。

表 3.2 收集 WHOIS 注册信息描述

WHOIS 注册信息	描述
Creation date	注册时间
Updated date	操作更新时间
Expiration date	到期时间
Registrar	注册商
其他 WHOIS 注册信息	域名注册人、域名所在地等

[4] http://s3-us-west-1.amazonaws.com/umbrella-static/index.html

[5] https://proprivacy.com/privacy-news/covid-19-malicious-domain-report

最后，将得到的数据集 D，按照 3:1:1 的比例划分训练集、测试集和验证集，并对应分配标签，最终域名数据集如表 3.3 所示。

表 3.3 域名文本和 WHOIS 注册信息数据集

类别	数量	训练集	验证集	测试集
合法	30000	24000	6000	6000
恶意	24000	14400	4800	4800

3.3.2 实验环境及评价指标

实验操作系统为 Ubuntu18.04，硬件环境为 Tesla V100 32G，开发语言为 Python3.6，采用深度学习框架 Tensorflow2.6 和 Keras2.6 实现模型搭建、训练与测试。

实验采用准确率(Accuracy)、精确率(Precision)、召回率(Recall)和 F1 值(F1-score)四项指标，以衡量模型性能，详细见 2.5 节恶意域名检测模型评价标准。

3.3.3 模型参数设置

实验过程中，合适的参数能够对模型训练和分类的结果产生积极影响[62]。本实验中，部分参数根据以往的经验设置，模型参数设置如表 3.4 所示。

表 3.4 模型参数设置

模型参数	值
Transformer 编码器层数	2
丢弃率	0.5
训练轮次	20
初始学习率	0.001
优化算法	Adam
Loss 函数	binary cross entropy

3.3.4 对比实验

（1）与基准模型的对比

本节将本章模型与基准模型 CNN 和 Transformer 在 3.3.1 节中数据集上进行实验对比，旨在验证所提出结合 CNN 和 Transformer 模型的有效性。在实验设置中，CNN 的卷积核大小设为 2+4，注意力头数等参数与 Transformer 模型保持一致。实验结果如表 3.5 所示。分析可知，本章模型在四项评价指标上均优于 CNN 和 Transformer 基准模型，从而验证了融合两者的有效性。相较于单一使用 CNN 或 Transformer 模型，本章模型结合 Transformer 模型中注意力机制对长距离上下文关系建模能力和 CNN 模型提取局部细节特征的优势，从而，更充分建

模域名文本和注册信息语义表示，以获得较好的模型性能。虽然将 Transformer 和 CNN 所提取特征融合后本章模型特征维度增加，导致模型训练时间相较于 CNN 模型增加了 3 分钟，但模型性能的显著提升与时间成本的增加在合理可接受的范围内。

（2）与现有方法的对比

本节在 3.3.1 节中数据集上构建现有主流恶意域名检测模型 GoogleNet V1[18]、DBD[21]、LSTM[38]和 AB-BiLSTM[63]，与本章模型实验比较。对比实验结果如表 3.5 所示。

表 3.5 对比实验结果

模型	Accuracy	Precision	Recall	F1-score	训练时间(分钟)
CNN	0.9837	0.9846	0.9778	0.9812	**10**
Transformer	0.9872	0.9872	0.9801	0.9825	12
GoogleNet V1	0.9464	09510	0.9655	0.9470	14
LSTM	0.9721	0.9673	0.9651	0.9656	15
DBD	0.9803	0.9762	0.9607	0.9700	16
AB-BiLSTM	0.9870	0.9834	0.9816	0.9820	20
本章模型	**0.9924**	**0.9942**	**0.9881**	**0.9896**	13

分析可知，本章模型在四项指标均优于其他现有主流模型，准确率和精确率高达 99.24%和 99.42%，相较于现有最佳模型 AB-BiLSTM，分别提升了 0.54%和 1.08%，证明本章模型能够有效检测恶意域名。其中，GoogleNet V1 模型采用 Inception 模块，并行多尺度的卷积核和池化操作，以提高模型捕获不同尺度信息的能力，然而，相较于本章模型，GoogleNet V1 模型架构复杂，参数量较大，导致训练耗时较长。此外，针对恶意域名检测任务，GoogleNet V1 采用较大尺寸的卷积核，导致模型无法捕获域名字符间组合关系，因此效果较差。LSTM 模型能够有效捕获域名上下文依赖关系，但对于域名的局部细节表征能力不足，且计算量较大，导致训练时间相对较长。DBD 模型采用 CNN 和 LSTM 以检测恶意域名，利用 CNN 和 LSTM 的优势，能够有效捕获域名的局部和全局信息。然而，该模型参数较多，结构较复杂，训练时间较长。AB-BiLSTM 模型结合 BiLSTM 模型的双向序列建模能力和注意力机制对于关键信息的加权学习优势，能够更有效地捕获域名的上下文依赖关系并关注域名中重要部分，效果仅次于本章模型。但是 BiLSTM 无法并行计算，且注意力机制应用在 BiLSTM 层之后，进一步增加了训练的时间成本，因此训练所需时间最长。

实验结果表明，本章模型结合 Transformer 模型中注意力机制和 CNN 模型的优势，能够更充分地捕获域名的全局语义表示和局部细节特征，相较于现有主

流恶意域名检测模型，本章模型性能最佳。此外，Transformer 和 CNN 模型计算步骤可以独立执行，因此可以在多个计算单元上同时进行处理，因此，相较于现有主流恶意域名检测模型，本章模型训练时间开销最小。

模型测试阶段包含 10800 条域名，预测结果如图 3.3 所示。其中，正确预测的有 10719 条，错误预测的共有 81 条，分析原因为本章结合域名文本和注册特征，不一定能够充分反映恶意域名的特征。可能存在一些重要的特征未被考虑或者没有充分利用，可以尝试添加更多有助于区分恶意域名和正常域名的特征。

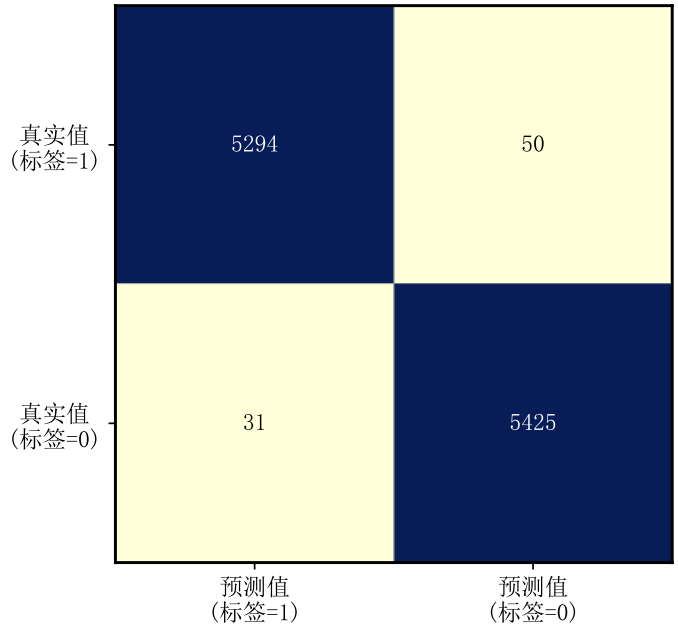


图 3.3 本章模型测试结果的混淆矩阵

3.3.5 消融实验

（1）各模块对于模型性能影响

为了验证各模块及 WHOIS 注册信息对于模型性能的影响，将数据集按是否包含 WHOIS 注册信息划分为两部分，分别训练测试，结果如表 3.6 所示。

表 3.6 消融实验结果

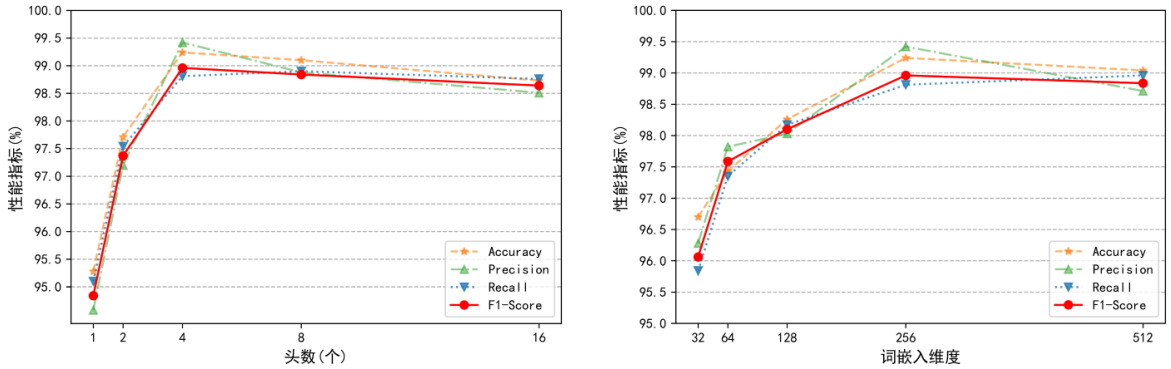
模型	WHOIS 注册信息	Accuracy	Precision	Recall	F1-score
Transformer	✓	0.9872	0.9872	0.9801	0.9825
	×	0.9781	0.9614	0.9808	0.9705
CNN	✓	0.9837	0.9846	0.9778	0.9812
	×	0.9810	0.9687	0.9745	0.9716
本章模型	✓	**0.9924**	**0.9942**	**0.9881**	**0.9896**
	×	0.9851	0.9655	0.9817	0.9730

由表 3.6 可以看出，首先，域名注册特征对于模型性能影响较大，使用包含 WHOIS 注册信息的数据集相较于不包含 WHOIS 注册信息的数据集在

Transformer 模型上，准确率提升了 0.9%，精确率提升了 2.58%，F1 值提升了 1.2%，表明 WHOIS 注册信息能够丰富域名表达，有效提高恶意域名检测精度。其次，相较于单独采用 Transformer 和 CNN 模型，本章提出的模型在四个评价指标上均获得了最优结果，表明结合 Transformer 和 CNN 模型能够更充分地捕获域名文本和注册信息语义表示。

（2）注意力头数与嵌入维度分析

不同的词嵌入维度和 Transformer 中注意力头的个数能够较大影响模型性能，为研究词嵌入维度和注意力头数对于模型性能的影响，分别设置常见的不同大小的注意力头数和词嵌入维度进行实验，结果如图 3.4 所示。



(a) 不同注意力头数模型性能比较　　(b) 不同词嵌入维度模型性能比较

图 3.4 不同注意力头数和词嵌入维度模型性能比较

随着嵌入维度的增大和头数的增加，模型性能显著提高，但当注意力头数超过 4，词嵌入维度超过 256 后模型提升较小，甚至出现下降趋势。说明过大的注意力头数和词嵌入维度导致模型参数量过大，模型容易过拟合。故实验最终将嵌入维度设置为 256，头数设置为 4。

（3）卷积核分析

为了研究不同卷积核大小组合对模型性能的影响，选取不同大小的卷积核组合进行实验并分析，具体实验结果如表 3.7 所示。

表 3.7 不同卷积核大小模型性能比较

卷积核	Accuracy	Precision	Recall	F1-score
2	0.9654	0.9702	0.9635	0.9667
3	0.9703	0.9724	0.9719	0.9722
4	0.9767	0.9781	0.9766	0.9774
2+3	0.9841	0.9872	0.9813	0.9843
2+4	**0.9924**	**0.9942**	**0.9881**	**0.9896**
3+4	0.9834	0.9805	0.9851	0.9828

结果表明，当仅使用单一尺寸的卷积核时，模型无法充分捕获域名的局部语义信息，导致效果较差。然而，当结合不同尺寸的卷积核时，模型能够捕获不同

尺度的域名语义信息，性能显著提升。特别是当结合卷积核大小分别为 2 和 4 时，模型性能达到最佳，这验证了英文单词通常由 2 至 4 个字符构成的构词特点。因此，实验参数最终设置中，卷积模块选择卷积核尺寸分别为 2 和 4 的组合。

3.3.6 泛化性实验

为了进一步验证本章模型的泛化性，在 CICBellDNS2021 数据集上对其进行了评估，该数据集是由加拿大网络安全研究所和网络威胁情报中心的合作项目生成并发布的大型 DNS 数据集[6]。首先，随机选取 5000 个恶意和良性域名并获取域名 WHOIS 注册信息；然后，采用本章模型和对比实验中现有恶意域名检测模型效果较好的 AB-BiLSTM 提取域名文本和注册信息特征，从而实现域名分类；最后，实验结果如表 3.8 所示。分析表明，本章模型在 CICBellDNS2021 数据集上的准确率和精确率均超过 95%。相较于现有最佳模型 AB-BiLSTM，召回率提高了 2.02%，F1 值提高了 0.98%。进一步证明本章模型具有较强的泛化性，可以有效地检测出不同类型的恶意域名。

表 3.8 CICBellDNS2021 数据集评估结果

模型	Accuracy	Precision	Recall	F1-score
AB-BiLSTM	0.9432	0.9515	0.9170	0.9329
本章模型	**0.9505**	**0.9601**	**0.9372**	**0.9427**

3.4 小结

本章提出一种融合域名文本和注册特征的恶意域名检测方法。获取域名 WHOIS 注册信息，构建域名注册特征表示，以丰富域名特征，并忽略部分对模型影响较小的 WHOIS 注册信息，增强模型泛化能力。此外，结合 Transformer 和 CNN 模型提取域名文本和注册信息的全局和局部特征。在新冠疫情相关域名数据集上实验，结果表明，WHOIS 注册信息能够有效提高恶意域名检测效果，本章模型相较于现有主流模型，效果较好且检测效率较高。此外，在 CICBellDNS2021 数据集上实验，结果表明，本章模型泛化性较好，能够有效检测不同类型恶意域名。

[6] https://www.unb.ca/cic/datasets/dns-2021.html

第4章 融合域名解析特征的恶意域名检测

4.1 引言

本章在第三章融合域名注册特征的基础上，引入域名 DNS 解析记录，以构建域名解析特征，并将其与域名文本、注册特征融合，从而丰富域名行为模式特征表示。此外，在第三章结合 Transformer 和 CNN 模型的基础上，设计了一种恶意域名检测模型 Iconformer，结合注意力机制的全局建模和 CNN 的局部建模优点，并引入马卡龙式前馈模块提取多级特征，使模型更充分地学习域名的潜在特征表示。

其中，MHPSSA 模块能够缓解传统 Transformer 中 MHSA 存在的稀疏性的问题，并有效降低模型计算复杂度和参数量。此外，卷积模块结合蒸馏操作，突出重要注意力权重，减少冗余信息。将 MHPSSA 模块与卷积模块相结合，从而有效减小模型参数量，提高模型泛化性。

4.2 模型结构

融合域名解析特征的恶意域名检测模型结构如图 4.1 所示，模型包含输入层、特征融合层、特征提取层和输出层。首先，输入层将域名字符串、WHOIS 注册信息和 DNS 解析记录预处理后转换为能被深度学习模型学习的特征向量；其次，特征融合层采用 AFF 模块融合域名文本、注册和解析三种特征；然后，特征提取层采用 Iconformer 提取域名特征；最后，输出层将域名特征向量经过激活函数为 Sigmoid 的全连接层，得到域名分类结果 y，实现恶意域名检测。

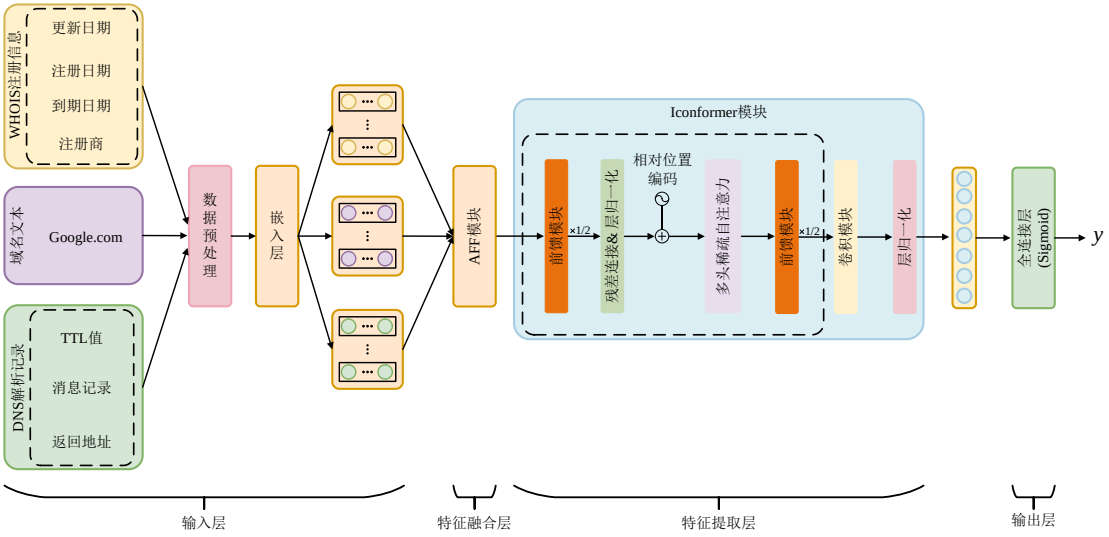


图 4.1 融合域名解析特征的恶意域名检测模型结构

4.2.1 输入层

本层输入为域名字符串、WHOIS 注册信息、DNS 解析记录，其中 WHOIS 注册信息包含域名注册时间、更新时间、到期时间和注册商，DNS 解析记录包含域名 IP 地址、消息记录和 TTL 值。首先，过滤数据集中重复的域名字符串、WHOIS 注册信息和 DNS 解析记录，并将域名字符串中大写字母转换为小写。

其次，将输入域名、WHOIS 和 DNS 字符串长度分别统一为 L_1、L_2 和 L_3。对于长度不足的域名、WHOIS 和 DNS 字符串，在字符串前端填充“0”字符。

然后，分别统计数据集中域名字符串的字母、数字、特殊字符，WHOIS 注册信息中时间、注册商，DNS 解析记录中 IP 地址段、消息记录和 TTL 值，从而构建域名字符串字典 D_1、WHOIS 注册信息字典 D_2 和 DNS 解析记录字典 D_3。

最后，采用 Word2Vec 方法作为词嵌入方法，将字典转化为能够被深度学习模型学习的特征向量表示。

4.2.2 特征融合层

鉴于输入层用于字符串长度填充的“0”字符对特征提取造成的干扰，采用 AFF 模块[64]融合域名文本、注册和解析特征。AFF 通过通道注意力学习每种特征的表达，能够有效地融合不同尺度的特征，此外，注意力机制动态地为不同位置分配权重，增强域名特征提取过程中抗干扰能力。AFF 模块结构如图 4.2 所示。

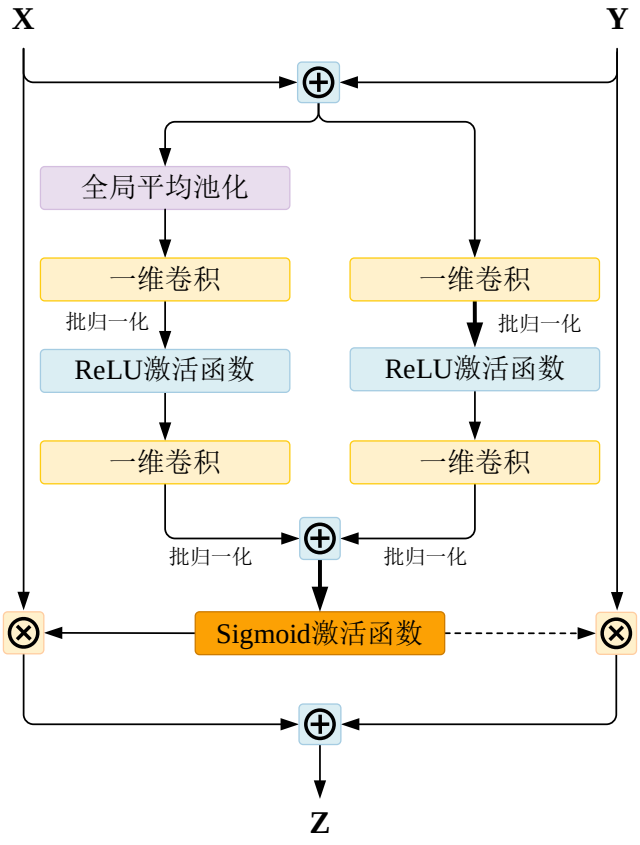


图 4.2 AFF 结构

假设域名文本、注册、解析特征向量分别为 X_1， X_2， X_3，经过特征融合层得到域名融合特征向量 Y，计算公式如式(4.1)所示。

$$Y = \mathrm{AFF}\left(\mathrm{AFF}\left(X_1, X_2\right), X_3\right) \tag{4.1}$$

4.2.3 特征提取层

近年来，基于多头自注意力的 Transformer 模型由于其捕获长距离依赖的特性而在 NLP 中得到了广泛采用。此外，CNN 能够有效捕获上下文局部特征，因而在恶意域名检测[19,65]也取得了成功。在恶意域名检测任务中，研究者也尝试结合两者的优点。Transformer[23]最初由一个编码器和一个解码器组成，其目的是建立一个端到端的机器翻译模型，对于恶意域名检测任务，只使用编码器来提取有效特征。基于 Transformer 模型，一种改进为 Conformer[66]，在编码器中加入卷积模块以学习局部特征，并在自动语音识别任务获得了较好的效果，另一种改进为 Informer[67]，提出 MHPSSA 模块，将时间和空间复杂度从 Transformer 的 $O\left(L^2\right)$ 降低到 $O\left(L\log L\right)$。因此，本章设计了 Iconformer 模型，融合 Conformer 和 Informer 的设计理念，兼具 CNN 和 Transformer 的优势，更充分地提取域名的序列特征，从而提高检测效果。

Iconformer 编码器结构如图 4.3 所示，由两个模块组成，即 MHPSSA 模块和卷积模块，其中 MHPSSA 模块包含两个半步前馈模块及 MHPSSA。

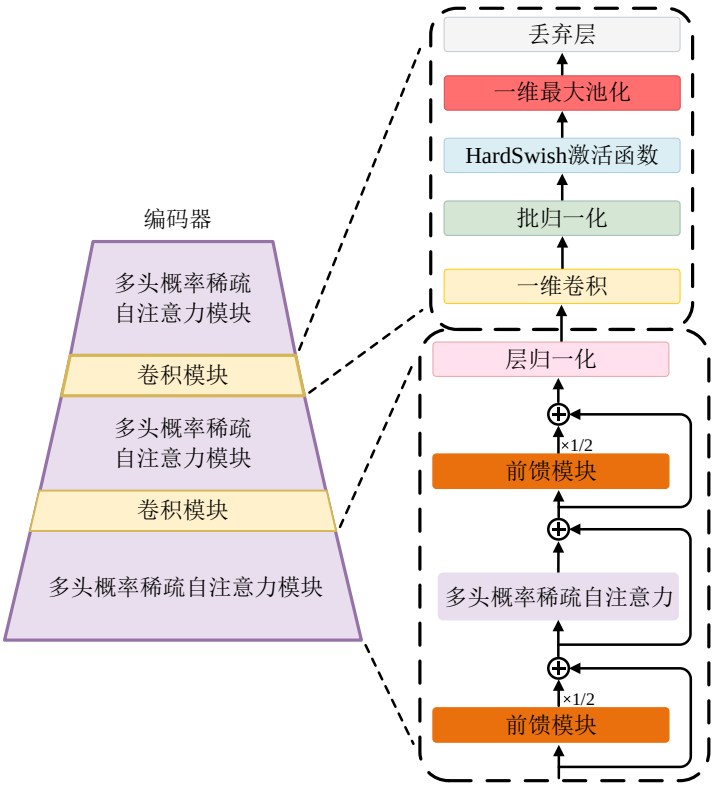


图 4.3 Iconformer 编码器结构

（1）MHPSSA 模块

传统的 MHSA 是基于输入元组来定义的，即 Q(query)、K(key)和 V(value)，进行缩放点积运算得到 $\mathrm{A}(Q,K,V)=\mathrm{Softmax}(QK^{\mathrm{T}}/\sqrt{d})V$，其中 $Q\in\mathbb{R}^{L_Q\times d}$，$K\in\mathbb{R}^{L_K\times d}$，$V\in\mathbb{R}^{L_V\times d}$，$d$ 为输入维度。近来，部分研究者发现传统的自注意力权重的概率分布具有稀疏性，自注意力权重分数呈长尾分布，即少数点积对贡献了主要注意力，其他点积对仅产生次要注意力，可以忽略。

因此，本节采用 MHPSSA 提取域名全局特征，结合来自 Transformer-XL[68] 的相对位置编码方案。相对位置编码考虑了位置之间的相对关系，能够捕捉到更复杂的位置关系，更有效泛化不同输入长度，增强编码器因域名长度变化的鲁棒性。此外，采用 Pre-Norm 残差单元[69,70]，以训练和正则化更深层次的模型。MHPSSA 模块结构如图 4.4 所示。

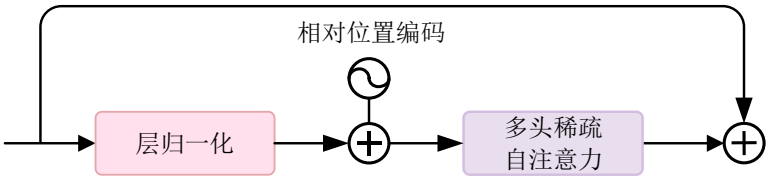


图 4.4 MHPSSA 模块结构

MHPSSA 如算法 4.1 所示。

算法 4.1 MHPSSA

输入： 张量 $Q\in\mathbb{R}^{L_Q\times d}$，$K\in\mathbb{R}^{L_K\times d}$，$V\in\mathbb{R}^{L_V\times d}$

输出： 自注意力特征映射 S

1. 设置超参数 $c,u=c\ln m$ 和 $U=m\ln n$
2. 从 K 中随机选择 U 个 K 记作 $\bar{K}$
3. 设置采样分数 $\bar{S}=Q\bar{K}^{\mathrm{T}}$
4. 计算 $M=\max(\bar{S})\text{-}\mathrm{mean}(\bar{S})$
5. 将分数低于 M 的前 u 个 Q 设置为 $\bar{Q}$
6. 计算 $S_1=\mathrm{softmax}(\bar{Q}\bar{K}^{\mathrm{T}}/\sqrt{d})\cdot V$
7. 设置 $S_0=\mathrm{mean}(V)$
8. 根据初始对应设置 $S=\{S_1,S_0\}$
9. 返回 S

MHPSSA 首先对 K 进行采样得到 $\bar{K}$，然后，计算每个 $q_i\in Q$ 与每个 $\bar{K}$ 的稀疏性度量 M，计算公式如式(4.2)所示。

$$M(q_i,K)=\max_j\left\{\frac{q_ik_j^{\mathrm{T}}}{\sqrt{d}}\right\}-\frac{1}{L_K}\sum_{j=1}^{L_K}\frac{q_ik_j^{\mathrm{T}}}{\sqrt{d}} \tag{4.2}$$

其中，$q_i\in\mathbb{R}^d$，$k_j\in\mathbb{R}^d$，分别为 Q 和 $\bar{K}$ 的第 i、j 行

最后，选取稀疏性度量最大的 u 个 Q 记作 $\bar{Q}$，并与每个 K 计算点积以获得

MHPSSA 分数，计算公式如式(4.3)所示，其余的Q取输入均值。

$$A(Q,K,V)=\mathrm{Softmax}(\frac{\bar{Q}K^{\mathrm{T}}}{\sqrt{d}})V \tag{4.3}$$

其中，$\bar{Q}$是与q大小相同的稀疏矩阵。

（2）卷积模块

受 Informer[67]的启发，Iconformer 中引入卷积模块以弥补其在提取局部特征方面的不足，并采用 HardSwish 激活函数[71]加快模型收敛速度。首先，卷积模块能够有效学习域名局部语义特征表示；其次，MHPSSA 中存在 value 的冗余组合，Iconformer 采用蒸馏操作以突出主导的注意力权重，减少冗余信息。同时，通过聚焦的自注意力特征，能够有效地在下一层捕获序列中的关键信息。蒸馏过程从第 j 层向前到第 $j+1$层计算过程如式(4.4)所示。

$$X_{j+1}=\mathrm{MaxPool}\left(\mathrm{HardSwish}\left(\mathrm{Conv1d}\left(\left[X_j\right]_{AB}\right)\right)\right) \tag{4.4}$$

卷积模块如图 4.5 所示。首先经过卷积核尺寸为 2 和 4 的一维卷积层；其次，采用 HardSwish 激活函数，以加快模型收敛速度，防止模型过拟合；最后，采用步长为 2 的最大池化层以突出重要注意力权重，并经过丢弃层输出。

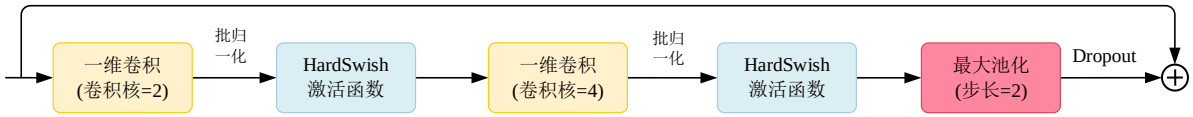


图 4.5 卷积模块结构

（3）前馈模块

传统的 Transformer[23]中在 MHSA 层之后采用前馈模块，首先经过两个线性层和非线性激活函数，然后经过残差连接和层归一化。相较于 Transformer 中的单层前馈模块，Iconformer 采用一对马卡龙(Macaron)式前馈模块[72]，即将原始 Transformer 中的前馈层分为两个半步(half-step)前馈子层，前半步在 MHPSSA 前对输入进行预处理，后半步进一步提取 MHPSSA 的输出特征。马卡龙式前馈模块通过多级特征提取能够使模型更有效地学习输入域名序列的特征表示。

前馈模块如图 4.6 所示。其中，Pre-Norm 残差单元能够提升训练稳定性并加速模型的收敛。此外，采用 HardSwish 激活函数，以加快模型收敛速度，防止模型过拟合。

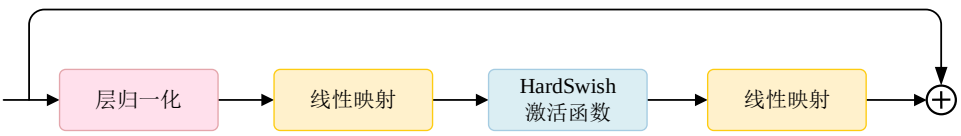


图 4.6 前馈模块结构

（4）Iconformer 编码器

Iconformer 编码器如图 4.3 所示，包含 MHPSSA 模块和卷积模块。特征融合层得到的域名特征向量经过 MHPSSA 模块，其中包含两个半步前馈模块，将 MHPSSA 夹在中间。其次，经过卷积模块，包含卷积核分别为 2 和 4 的两个一维卷积层，最后，通过最大池化层将特征降维后输出。对于 Iconformer 的输入 x，编码器输出 y，计算过程如式(4.5)-(4.8)所示。

$$\tilde{x} = x + \frac{1}{2}\mathrm{FFN}(x) \tag{4.5}$$

$$x' = \tilde{x} + \mathrm{MHPSSA}(\tilde{x}) \tag{4.6}$$

$$x'' = \mathrm{LayerNorm}\left(x' + \frac{1}{2}\mathrm{FFN}(x')\right) \tag{4.7}$$

$$y = \mathrm{Conv}(x'') \tag{4.8}$$

其中，FFN 为前馈模块，MHPSSA 为多头概率稀疏自注意力模块，Conv 为第（2）小节中描述的卷积模块。

Iconformer结构类似金字塔结构，能够有效提取域名全局及局部特征，提高恶意域名检测精度；并减少模型参数量，以降低模型训练推理的时间和内存开销；同时，增强模型训练稳定性，提高模型泛化能力。Iconformer的上述优势归结于以下贡献点：

1) Iconformer 采用 MHPSSA 并结合相对位置编码。其中 MHPSSA 通过稀疏性度量选取重要的注意力点积对，能够有效减少模型参数量，以降低模型训练推理的时间和内存开销。并结合相对位置编码，有效泛化不同长度输入序列。第 4.3.6 节消融实验中将 Iconformer 与部分主流的自注意力模块进行比较，实验结果得出，Iconformer 参数量较少，且检测精度较高。

2) Iconformer 采用马卡龙式前馈模块。第 4.3.6 节消融实验中比较了马卡龙式半步前馈模块、传统的单层前馈模块和全步前馈模块。实验结果得出，相较于单层前馈模块或全步前馈模块，Iconformer中两个半步前馈模块能够显著提高模型性能。

3) Iconformer采用 HardSwish 激活函数和 Pre-Norm 残差单元，以加快模型训练推理速度，增强模型训练稳定性。第 4.3.6 节消融实验中比较了将 HardSwish 替换为常用的 ReLU 激活函数和将 Pre-Norm 残差单元替换为传统 Post-Norm 残差单元。实验结果得出，HardSwish 激活函数和 Pre-Norm 残差单元具有较好的性能。

4) Iconformer 采用蒸馏卷积模块。卷积模块能够有效学习域名局部语义特征，并采用最大池化蒸馏操作，突出重要注意力权重，减少冗余信息。第 4.3.6 节消融实验中研究了卷积核的不同选择对于模型性能的影响。实验结果得出，组合卷积核大小为 2 和 4 的卷积模块能够更有效提取域名局部特征。

4.2.4 输出层

输出层首先将 Iconformer 得到的域名特征向量经过丢弃层，其次，采用激活函数为 Sigmoid 的全连接层以输出域名判别结果。

4.3 实验设计与结果分析

4.3.1 数据集

本节采用公开可获取的合法和恶意域名数据集作为基准，构建了一个包含域名文本、WHOIS 注册信息、DNS 解析记录的域名数据集。首先，从一个合法域名数据集和八个恶意域名数据集共收集到 100 万条合法域名及 740979 条恶意域名，其中恶意域名包含三类，恶意软件、钓鱼网站、垃圾邮件，基准数据集详细信息如表 4.1 所示；其次，采用 python-whois 工具和 dnspython 工具获取域名 WHOIS 注册信息及 DNS 解析记录，以构建域名的注册和解析特征。详细的获取流程将在第（2）小节中描述；最后，鉴于真实网络环境中合法域名的数量远大于恶意域名，并且数据集中部分域名已失效，故大多域名无法进行解析，因此，能够完整获取 WHOIS 注册信息和 DNS 解析记录的合法域名远大于恶意域名。此外，筛除获取到的域名数据集中空值、错误值，并删除部分域名中协议名及主机名，仅保留顶级域名及二级域名，最终数据集如表 4.3 所示。

（1）数据来源

公开的合法域名数据集包含基于流量统计的顶级网站列表 Amazon Alexa Top Sites[7]及 Cisco Umbrella Top 1 Million[8]等，它们通过不同的筛选机制发布全球范围内用户访问量排名前 100 万的域名。其中，Alexa 通过浏览器工具栏插件收集用户访问的网站。但是自 2021 年起，Alexa 不再提供最新的数据集。Cisco Umbrella 通过分析 DNS 流量，关注 DNS 解析记录中访问次数最多的域名。

真实网络环境中，合法域名的数量远大于恶意域名，直接随机采样收集恶意域名样本较难，因此，采用公开的恶意域名数据集作为基准数据集。其中网络垃圾邮件数据集 WEBSPAM[9]由米兰大学网络实验室发布，他们抓取并分析顶级域名为 uk 的域名。JwSpam[10]通过垃圾邮件过滤器 jwSpamSpy 提取垃圾邮件中的域名和电子邮件地址，并综合考虑域名注册时间和注册商等因素。

[7] https://www.alexa.com/topsites

[8] http://s3-us-west-1.amazonaws.com/umbrella-static/index.html

[9] https://chato.cl/webspam/datasets/

[10] https://joewein.net/bl-log/bl-log.htm

ISCX-URL2016[11]由加拿大网络安全研究所(The Canadian Institute for Cybersecurity, CIC)于 2016 年发布。CICBellDNS2021[12]由 CIC 和加拿大贝尔电信公司共同发布的大型 DNS 数据集，其中恶意域名样本涵盖垃圾邮件、网络钓鱼和恶意软件三类。

OpenPhish[13]分析来自世界各地各种来源的数百万个网址，并实时发布检测到新的网络钓鱼网址。PhishStats[14]收集网络钓鱼 URL 并与信息安全社区共享。PhishTank[15]由思科运营，任何人都可以提交、验证、跟踪和共享网络钓鱼数据。GitHub[16]采用 PyFunceble 测试工具验证所有已知网络钓鱼 URL 的状态，每小时发布一次数据集，表 4.1 为收集到的基准域名数据集。

表 4.1 基准域名数据集

数据集来源	数量	类别
Cisco Umbrella	100 万	Benign
WEBSPAM	2364	Spam
JwSpam	40285	Spam
CICBellDNS2021	30636	Malware
	17388	Phishing
	1126	Spam
ISCX-URL2016	2691	Malware
	9957	Phishing
	11921	Spam
OpenPhish	500	Phishing
PhishStats	4381	Phishing
PhishTank	30326	Phishing
GitHub	589404	Phishing

（2）数据收集

域名 WHOIS 注册信息主要包含注册日期、注册机构等信息，通过注册信息可以验证域名持有者身份，检查域名的可用性，解析域名的生存时间线，通常注册年限短、频繁变更信息的域名可疑度较高[73]。DNS 解析记录包含返回类型、

[11] https://www.unb.ca/cic/datasets/url-2016.html

[12] https://www.unb.ca/cic/datasets/dns-2021.html

[13] https://openphish.com/index.html

[14] https://phishstats.info

[15] https://www.phishtank.com/index.php

[16] https://github.com/mitchellkrogza/Phishing.Database

IP 地址和 TTL 等信息，反映域名指向的网络位置，解析结果中若出现恶意 IP 地址表明域名可能被攻击者控制。合法域名通常为提高其网站访问速度，TTL 值一般较大，而恶意域名因其需要经常改变其解析记录的特点，TTL 值一般较小，因此，分析注册和解析记录信息可以有效判别域名是否为恶意。本节沿用 3.3.1 中所收集的部分 WHOIS 注册信息，详细信息见表 3.2。本节所收集的部分 DNS 解析记录描述如表 4.2 所示。

表 4.2 收集 DNS 解析记录描述

DNS 解析记录	描述
IP	域名 IP 地址
RdType	域名返回记录类型
TTL	生存时间

采用 python 第三方库 python-whois 及 dnspython 获取表 4.1 中域名的部分 WHOIS 注册信息及 DNS 解析记录，以缩短数据获取时间，也能为后续的恶意域名检测缩短处理时间，相关代码详见附录 A。但是公开的恶意域名数据集中大多域名已经失效，无法获取 WHOIS 注册信息及 DNS 解析记录。为了保证数据的有效性和一致性，本节将数据集进行整理和标准化，包括筛除关键注册信息不全的域名样本，去除冗余数据，并为每个域名样本分配标签。最后，为保证数据集的平衡性，随机抽取部分合法域名，将数据集按照 3:1:1 的比例划分训练集、验证集和测试集，最终融合域名文本、注册和解析特征的域名数据集详细信息如表 4.3 所示。

表 4.3 域名文本、注册信息和解析记录数据集

类别	数量	训练集	验证集	测试集
合法	120000	72000	24000	24000
恶意	119250	71550	23850	23850

4.3.2 实验环境及评价指标

实验平台为 Ubuntu 20.10 操作系统和 NVIDIA Tesla A100 40G GPU。在 Python3.8.18 开发语言中采用 tensorflow-gpu2.6.0 和 Keras2.6.0 实现 Iconformer 模型的搭建、训练与测试。

本章沿用 3.3.2 小节所提准确率(Accuracy)、精确率(Precision)、召回率(Recall)和 F1 值(F1-score)评价指标，以衡量模型性能。

4.3.3 模型参数设置

Iconformer 模型训练过程中 epoch 设置为 20，采用 Adam 优化算法，batch size 设置为 64，丢弃率设置为 0.5，学习率大小设置为 0.0001，MHPSSA 中前馈模块线性层维度设置为 2048，注意力头数设置为 32。

因恶意域名检测为二分类任务，故采用 binary cross entropy 损失函数。域名字符串最大长度 L_1 为 120，WHOIS 注册信息最大长度 L_2 为 30，DNS 解析记录 L_3 最大长度为 40，域名字符串字典 D_1 大小为 39，WHOIS 注册信息字典 D_2 大小为 278，DNS 解析记录字典 D_3 大小为 263，词嵌入维度设置为 512。

4.3.4 对比实验

（1）与基准模型的对比

本节将 Iconformer 与基准模型 Informer、Conformer 进行比较，以验证本章所提出的改进的有效性。其中，Informer 和 Conformer 中卷积核大小设置为 2+4，注意力头数、丢弃率和词嵌入维度等参数与 Iconformer 设置一致。实验结果如表 4.4 所示。

分析可知，Iconformer 效果最佳，Informer 和 Conformer 较差，说明本章所提出的改进能够有效提升模型性能。相较于 Conformer，Iconformer 的准确率和 F1 值分别提升了 1.5%和 1%。首先，Iconformer 中卷积模块能够有效提取域名局部特征，并采用蒸馏操作，从而有效减少参数量的同时，突出输入序列中较为重要的注意力权重。而 Conformer 中卷积模块仅用作提取域名局部特征，且参数量较大；其次，Conformer 中采用传统的 MHSA 提取域名全局特征，而 Iconformer 采用 MHPSSA，能够缓解传统 MHSA 稀疏性的问题，降低模型复杂度，减少模型参数量；最后，Iconformer 将 Conformer 中 Swish 激活函数替换为 HardSwish 激活函数，能够增强模型训练稳定性且参数量较小。

相较于 Informer，Iconformer 的准确率和 F1 值分别提升了 1.25%和 0.95%。首先，Iconformer 结合相对位置编码，有效关注域名字符之间、注册信息之间、解析记录之间的相对位置关系，而 Informer 中未结合位置编码，导致模型无法有效学习输入序列相对位置关系，性能较弱；其次，Iconformer 采用双层半步连接马卡龙式 FFN，提取多级特征使模型更有效地学习输入域名序列的特征表示；最后，相较于 Informer，Iconformer 参数量较大，但是 Iconformer 获得了更好的性能，且参数量相差仅 5M。

（2）与现有方法的对比

本节将 Iconformer 模型与其他先进的 9 种模型进行了比较。实验结果如表 4.4 所示。

URLNet[19]采用字符级 CNN 和词级 CNN 以检测恶意域名，能够有效提取域名字符级和词级特征，但是无法捕获域名字符串、WHOIS、DNS 的全局依赖关系，所以效果较差。VAE-DNN[74]采用变分自编码器(Variational Autoencoders, VAE)提取域名特征，然后采用深度神经模型(Deep Neural Network, DNN)分类是否为恶意域名，通常，VAE 在数据量较小或数据具有连续变化的情况下表现

良好，但是，当输入为域名字符串、注册信息和解析记录时，数据量较大且离散，故效果较差。

表 4.4 对比实验结果

模型	参数量(M)	Accuracy	Precision	Recall	F1-score
Informer	183.8	0.9833	0.9850	0.9862	0.9856
Conformer	255.3	0.9808	0.9833	0.9869	0.9851
URLTran BERT	308.6	0.9834	0.9893	0.9811	0.9852
ITransformer_CNN	288.7	0.9761	0.9730	0.9810	0.9770
HMTransformer	294.1	0.9755	0.9804	0.9782	0.9793
D3-SACNN	282.2	0.9735	0.9778	0.9798	0.9788
N-Trans	201.5	0.9763	0.9788	0.9738	0.9763
Ruitong Liu	369.2	0.9841	0.9892	0.9856	0.9874
VAE-DNN	**80.7**	0.9457	0.9509	0.9537	0.9523
URLNet	103.9	0.9732	0.9722	0.9684	0.9703
Longwen Zhang	285.3	0.9811	0.9787	0.9827	0.9807
Iconformer	188.8	**0.9958**	**0.9986**	**0.9916**	**0.9951**

D3-SACNN[25]将 Transformer 与 CNN 结合以检测恶意域名。ITransformer_CNN[55]将输入域名通过词嵌入和 One-hot 编码后，结合 Transformer 和 CNN 模型检测恶意域名，但是输入为域名字符串、WHOIS 注册信息和 DNS 解析记录时，One-hot 编码存在特征维度高、稀疏性大的问题，容易导致梯度爆炸。HMTransformer[75]结合字符级词嵌入和 2-gram 词嵌入，然后采用 Transformer 和 CNN 模型以检测恶意域名，能够有效提取域名全局特征和局部特征，故效果优于 D3-SACNN 和 ITransformer_CNN。但是仅采用 2-gram 词嵌入无法完整捕获构词较复杂域名语义信息，存在部分特征丢失的问题。

N-Trans[26]采用 N-gram 算法处理恶意域名数据集，然后采用 Transformer 模型以检测恶意域名，但是，N-gram 算法导致模型参数量过大，增加训练推理的时间、空间开销。Zhang 等人[28]结合 Transformer 和多滤波器文本 CNN 检测恶意域名，能够有效捕获域名全局和局部特征，但是多滤波器文本 CNN 会导致模型参数量较大。

URLTran[24]采用基于 Transformer 架构的 BERT 和 RoBERTa 预训练模型，本节选取 BERT 预训练模型进行对比。Liu 等人[27]采用 CharBERT 预训练模型获取域名词嵌入矩阵，采用金字塔结构分层提取域名特征，其中 CharBERT 预训练模型能够获取较好的词嵌入矩阵，且金字塔结构分层模型能够有效提取域名特征，从而获得了相较于其他方法最好的性能。分析表 4.4 可知，采用预训练模型能够有效提高恶意域名检测性能，但是预训练模型通常需要大量的计算资源，同时在

目标任务微调阶段需要较高的推理成本，本章提出的 Iconformer 能够在减少模型参数的同时获得较好的性能。

实验结果可知，Iconformer 模型的准确率和精确率分别为 99.58%和 99.86%，分别比现有最好模型 Ruitong Liu[27]提升了 1.17%和 0.94%。结果表明，本章提出的 Iconformer 模型采用 MHPSSA 提取域名全局特征，CNN 提取域名局部特征，能够有效提高恶意域名检测精度，并减少参数量，从而降低模型训练推理的时间和内存开销；同时，增强训练稳定性，提高模型泛化能力。

模型测试阶段包含 47850 条域名，预测结果如图 4.7 所示。其中，正确预测的有 47647 条，错误预测的共有 203 条，分析原因为本章结合域名文本、注册和解析特征，能够充分表示域名特征，但由于部分类型域名样本分布不平衡，模型可能会倾向于更多地预测为样本数量较多的类别。可以尝试采用过采样等方法来处理样本不平衡问题。

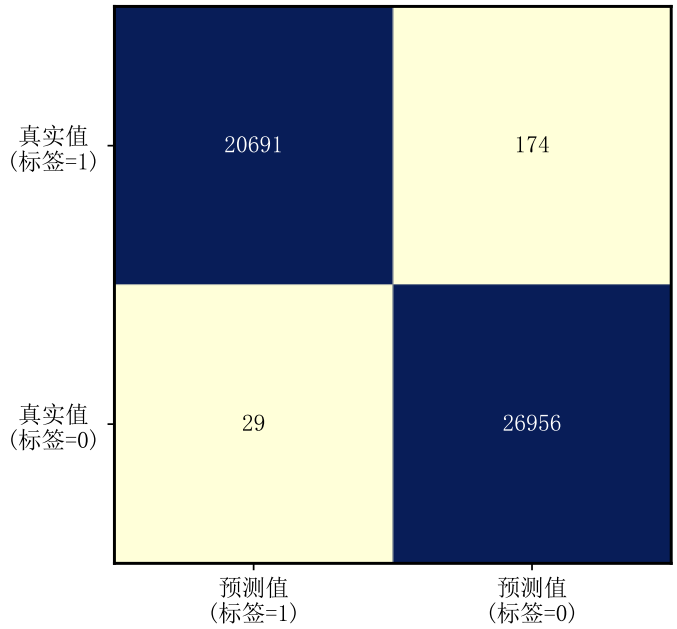


图 4.7 Iconformer 测试结果的混淆矩阵

（3）计算复杂度对比

表 4.5 为不同模型的计算复杂度，其中 L 为输入序列长度。分析可知，传统的 LSTM 时间复杂度和空间复杂度最低，但性能最弱。近来，大多研究者采用基于 Transformer 架构的模型，获得了较好的效果，但 Transformer 中，MHSA 将输入序列中的所有位置进行注意力计算，导致时间复杂度和空间复杂度均为 $O(L^2)$，之后，部分学者研究如何有效降低时间、空间消耗，Reformer 模型提出 LSH Self Attention，通过哈希机制仅选取离 Q 相近的 K 并作点积运算，从而降低计算复杂度。Sparse Attention 机制通过稀疏注意力矩阵，只计算部分 Q 和 K，有效减少计算复杂度。Iconformer 采用 MHPSSA，选取稀疏性度量最大的部分 Q 与 K 计算，将时间复杂度和空间复杂度降低到 $O(L\log L)$，同时检测性能较好。

表 4.5 不同模型计算复杂度

模型	时间复杂度	空间复杂度
LSTM	$O(L)$	$O(L)$
Transformer	$O(L^2)$	$O(L^2)$
Reformer	$O(L\log L)$	$O(L\log L)$
Sparse Attention	$O(L\sqrt{L})$	$O(L\sqrt{L})$
Iconformer	$O(L\log L)$	$O(L\log L)$

4.3.5 泛化性实验

（1）恶意域名数据集泛化性

为了探索 Iconformer 在不同类型域名数据集上的泛化性，首先，将 4.3.1 节中部分较少的恶意域名数据集组合，获得了 5 个临时恶意域名数据集，分别为 CICBellDNS2021、ISCX-URL2016、WEBSPAM+JwSpam 和 PhishTank+OpenPhish+Phishstats、GitHub，数据量分别为 9609、2197、4063、12622、90759。

其次，采用 python-whois 及 dnspython 获取 3.3.1 节中与新冠疫情相关的恶意域名的部分 WHOIS 注册信息及 DNS 解析记录，因部分域名已失效，最终共获取到 2 万条域名的完整注册信息和解析记录。

最终，构建了 6 个临时恶意域名数据集，并按照每个数据集 1:1 的比例随机抽取 4.3.1 节中合法域名数据集，数据集划分按照训练集/验证集/测试集(0.6/0.2/0.2)比例，最终构成临时数据集。实验结果如图 4.8 所示。

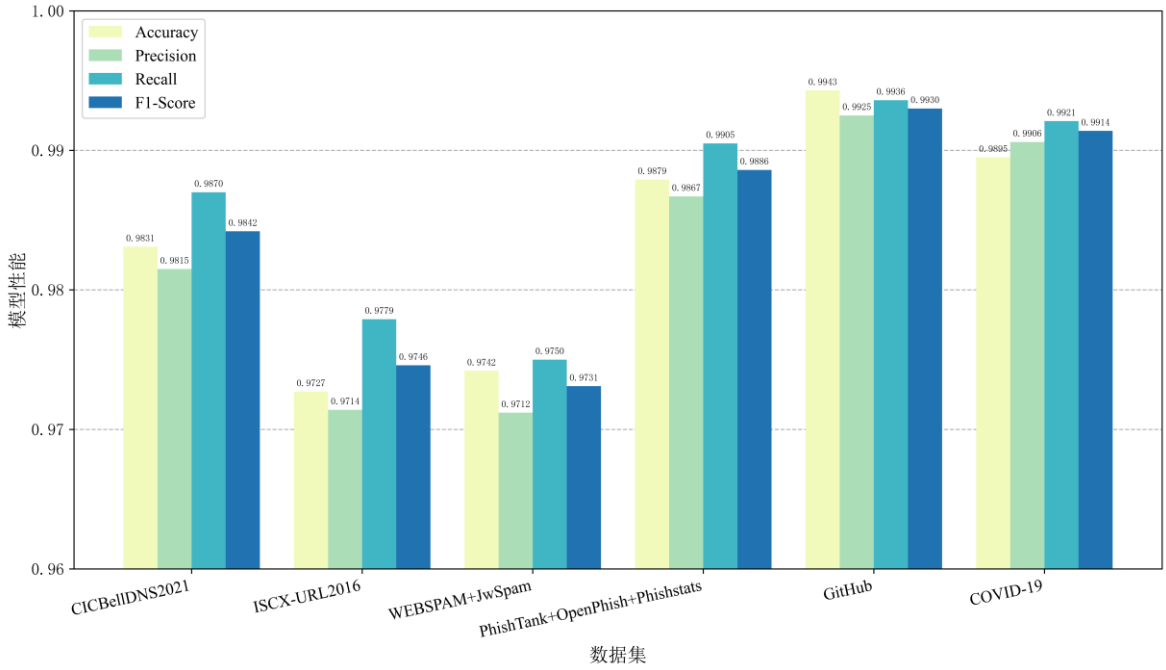


图 4.8 恶意域名数据集实验结果

分析得出，Iconformer 在不同类型数据集上四项评价指标均超过 97%，证

明模型具有较强的泛化性，其中，ISCX-URL2016 和 WEBSPAM+JwSpam 因数据量较小，导致模型无法完全拟合数据，故效果较差，而 CICBellDNS2021 和 PhishTank+OpenPhish+Phishstats 数据量较大，实验效果优于数据量较小的数据集，数据量最大的 GitHub 获得了最好的效果。此外，因新冠疫情(COVID-19)相关数据集数据量较大，也获得了较好的效果。

实验结果表明，结合域名解析特征能够有效丰富域名特征表达，提高恶意域名检测泛化能力，此外，Iconformer 采用 MHPSSA 模块，通过稀疏性度量选取重要的注意力点积，结合卷积模块蒸馏操作，有效减少模型参数量，从而提高模型泛化能力。

（2）DGA 数据集泛化性

为进一步探索 Iconformer 在不同类型域名数据集上的泛化性，在 360 安全实验室发布的大型 DGA 数据集[17]上对其进行了实验。首先，随机选取 10000 个 DGA 域名和良性域名，因 DGA 域名大多均失效，无法捕获 WHOIS 注册信息和 DNS 解析记录，因此，仅使用 DGA 域名文本以检测；然后，采用 Iconformer 和对比实验中效果最好的 Ruitong Liu[27]提取 DGA 域名文本特征，从而实现域名分类；最后，实验结果如表 4.6 所示。分析表明，虽然仅通过 DGA 域名文本，且数据集数量较少，但 Iconformer 的准确率和 F1 值均超过 98%，证明 Iconformer 在文本特征提取方面具有较强的泛化能力，能够在缺乏额外信息的情况下仍保持较好的分类性能。相较于 Ruitong Liu，准确率提高了 2.63%，F1 值提高了 2.38%，Ruitong Liu 模型采用金字塔结构以提取域名特征，模型参数量较大，导致泛化性较差。

表 4.6 DGA 数据集实验结果

模型	Accuracy	Precision	Recall	F1-score
Ruitong Liu	0.9620	0.9632	0.9671	0.9652
Iconformer	**0.9883**	**0.9874**	**0.9906**	**0.9890**

4.3.6 消融实验

（1）各模块对于模型性能影响

为了探索 Iconformer 中每个模块对于模型整体性能的影响，本节将部分模块替换或移除，1）将 HardSwish 激活函数更换为 ReLU；2）将 Pre-Norm 结构更换为 Post-Norm 结构；3）移除卷积模块；4）删除多头概率稀疏自注意力模块；5）用传统单层前馈模块替换马卡龙式前馈模块；6）移除相对位置编码。具体实验结果如表 4.7 所示。

[17] https://data.netlab.360.com/dga/

表 4.7 消融实验结果

模型结构	Accuracy	Precision	Recall	F1-score
Iconformer	**0.9958**	**0.9986**	**0.9916**	**0.9951**
−HardSwish-ReLU	0.9896	0.9938	0.9882	0.9910
−Pre-Norm-Post-Norm	0.9932	0.9982	0.9832	0.9907
−卷积模块	0.9663	0.9653	0.9621	0.9637
−MHPSSA 模块	0.9652	0.9596	0.9450	0.9523
−马卡龙式前馈模块	0.9878	0.9896	0.9868	0.9882
−相对位置编码	0.9795	0.9808	0.9712	0.9760

实验结果得出，卷积模块、MHPSSA 模块和相对位置编码对于 Iconformer 模型性能影响较大。MHPSSA 模块、卷积模块能够有效提取域名全局和局部语义特征，从而能够较大影响模型性能。其中，MHPSSA 模块能够有效提取域名文本、注册信息、解析信息的关联全局特征，并减少模型参数量，此外，结合相对位置编码，可以让模型学习字符之间的相对位置信息，并有效泛化不同长度输入序列。因此，当移除 MHPSSA 模块后，模型性能下降较大，且移除相对位置编码后，模型性能随之下降。

卷积模块能够有效提取域名文本字符组合特征、注册信息时间相关特征、解析记录组合特征，此外，采用蒸馏操作，进一步突出重要注意力权重，增强特征表达能力。HardSwish 激活函数和 Post-Norm 结构能够加速模型收敛和增强模型训练稳定性，实验结果表明将 HardSwish 和 Post-Norm 更换后，模型性能均有小幅度下降。马卡龙式前馈模块通过多级特征提取能够使模型更有效地学习域名序列的特征表示，故移除马卡龙式前馈模块后，模型性能随之下降。

（2）不同类型特征分析

本章采用 Iconformer 提取域名文本、注册和解析三种特征，为了探索不同特征对于模型整体性能的影响，逐步移除部分特征，具体实验结果如表 4.8 所示。结果得出，移除域名文本特征和解析特征后，模型性能下降较大。其中，恶意域名通常随机组合语义不相关的词汇，可读性较差，通过捕获域名字符串中字符间组合特征，能够有效识别域名是否为恶意，故当移除域名文本特征后，模型性能下降较大。其次，恶意域名通常具有特定的流量模式和行为特征，通过分析 DNS 流量可以识别这些恶意域名并采取相应的防御措施，故 DNS 解析记录对于恶意域名检测作用较大。此外，犯罪分子通常采用价格低廉的注册商以注册域名，并在发动攻击后不久就丢弃恶意域名，通过分析域名注册信息中的时间和注册商信息，可以有效识别恶意域名，故当移除注册信息特征后，模型性能随之下降。

表 4.8 不同特征模型性能比较

特征	Accuracy	Precision	Recall	F1-score
域名文本+注册信息+解析记录	**0.9958**	**0.9986**	**0.9916**	**0.9951**
−域名文本	0.9580	0.9665	0.9710	0.9687
−注册信息	0.9823	0.9806	0.9856	0.9831
−解析记录	0.9779	0.9728	0.9816	0.9772

（3）特征融合方法分析

为探究不同特征融合方法对于模型性能的影响，采用不同特征融合方法进行实验分析，具体实验结果如表 4.9 所示。

表 4.9 不同特征融合方法模型性能比较

方法	Accuracy	Precision	Recall	F1-score
Concatenate	0.9836	0.9803	0.9883	0.9843
Add	0.9850	0.9872	0.9888	0.9880
AFF	**0.9958**	**0.9986**	**0.9916**	**0.9951**

实验结果表明，相较于特征求和的特征融合方法，采用 AFF 模块融合域名文本、注册和解析特征，模型的准确率和精确率分别提升了 1.08%和 1.14%。基于特征组合的特征融合方法可以有效融合较浅层特征，但对于较复杂域名特征，融合效果较差。而 AFF 采用通道注意力，能够有效地融合域名文本、注册和解析中不同尺度的特征，动态地为不同位置输出分配权重，增强特征提取过程中抗干扰能力。

（4）注意力机制分析

为了探究不同的注意力机制对于模型性能的影响，采用传统的 MHSA 及几种先进的注意力机制实验后分析。具体实验结果如表 4.10 所示。

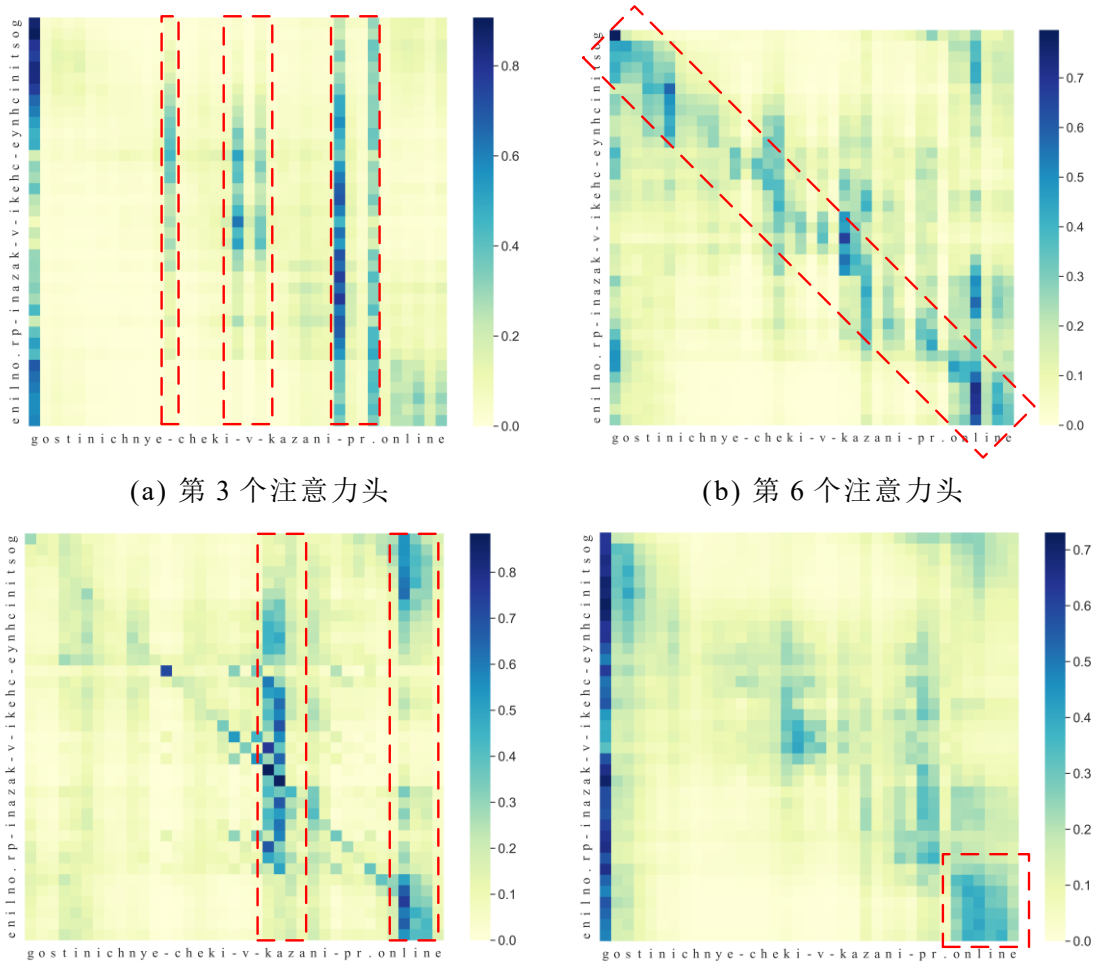
表 4.10 不同注意力机制模型性能比较

自注意力机制	参数量(M)	Accuracy	Precision	Recall	F1-score
MHSA	178.3	0.9788	0.9795	0.9815	0.9805
Sparse Attention	135.6	0.9884	0.9857	0.9911	0.9903
LSH Self Attention	127.4	0.9825	0.9836	0.9786	0.9811
MHPSSA	**108.8**	**0.9958**	**0.9986**	**0.9916**	**0.9951**

分析实验结果得出，相较于传统的 MHSA，经过改进的注意力机制可以较大幅度提升模型效果。首先，解决 Transformer 稀疏性问题的 Sparse Attention 机制，能够在减少模型参数量的同时效果均优于传统的 MHSA。其次，Reformer 模型提出局部敏感哈希(Locality sensitive hashing, LSH)自注意力机制，通过哈希机制选择离 Q 相近的 K，从而有效降低模型参数量，并获得了较好的效果。最后，MHPSSA 通过稀疏性度量选取重要的注意力点积，能够有效减

少模型参数量，以降低模型训练推理的时间和内存开销。MHPSSA 在各项指标上均明显优于其他注意力机制，尤其精确率指标较 Sparse Attention 提升 1.29%，且模型参数量相较于 LSH 自注意力机制降低了 14.6%。

为验证 MHPSSA 的有效性，选取恶意域名 gostinichnye-cheki-v-kazani-pr.online 作为 Iconformer 模型输入，将 MHPSSA 模块中第一层注意力权重矩阵结果输出并采用热力图进行可视化，部分注意力头的权值可视化结果如图 4.9 所示。观察多头注意力中不同头的注意力权重矩阵，可以发现不同的注意力头能够关注域名字符串不同层次的特征。图 4.9(a)中，第 3 个注意力头突出了域名中连接词(例如“-”和“.”)，在这些连接词出现的位置上有更高的注意力权重。图 4.9(b)中，第 6 个注意力头则捕捉字符之间的位置顺序信息，其对角线元素的权重较大，表示对域名中字符顺序关系的建模。图 4.9(c)中，第 9 个注意力头关注到词组语义信息，不同词组之间存在一定的注意力关联。图 4.9(d)中，第 12 个注意力头关注到词语语义关联，例如，“online”一词中的每个字符之间具有较强的注意力关联。



(a) 第 3 个注意力头　　(b) 第 6 个注意力头

(c) 第 9 个注意力头　　(d) 第 12 个注意力头

图 4.9 部分注意力头的权值可视化

（5）马卡龙式前馈模块分析

传统的 Transformer 采用单层前馈模块将多头自注意力层处理后的特征进行非线性变换和特征提取。为了探究不同结构的前馈模块对于模型性能的影响，分别将马卡龙式前馈模块、双层全步前馈模块和单层前馈模块进行实验分析，具体实验结果如表 4.11 所示。

表 4.11 不同前馈结构模型性能比较

结构	Accuracy	Precision	Recall	F1-score
单层前馈模块	0.9878	0.9896	0.9868	0.9882
全步前馈模块	0.9842	0.9858	0.9836	0.9847
马卡龙式前馈模块	**0.9958**	**0.9986**	**0.9916**	**0.9951**

实验结果表明，相较单层前馈模块，采用双层半步结构的马卡龙式前馈模块效果最佳，模型准确率、精确率分别提升了 0.8%和 0.9%。然而，双层结构的全步前馈模块的表现反而相较于单层前馈模块差，说明直接堆叠前馈层数不一定能够提升模型性能。

（6）卷积核分析

为探究不同卷积核大小对模型性能的影响，选取不同尺寸的卷积核组合进行实验后分析，实验结果如表 4.12 所示。

表 4.12 不同卷积核组合模型性能比较

卷积核	Accuracy	Precision	Recall	F1-score
2	0.9795	0.9824	0.9809	0.9816
3	0.9837	0.9831	0.9843	0.9837
4	0.9895	0.9882	0.9890	0.9886
2+3	0.9867	0.9904	0.9887	0.9895
2+4	**0.9958**	**0.9986**	**0.9916**	**0.9951**
3+4	0.9911	0.9887	0.9896	0.9890

实验结果显示，当采用单一卷积核时，尺寸为 2、3 和 4 时获得了较好的效果。鉴于英文单词通常由 2 至 4 个字符构成，因此较小的卷积核能够捕捉细节特征。然而，仅使用单一卷积核，无法有效捕获不同尺度的域名特征。在组合卷积核的实验中，结合小卷积核和大卷积核时表现最佳，如 2+4 和 2+3 的组合，表明结合不同大小的卷积核对于多尺度特征学习至关重要，当卷积核组合为 2+4 时，效果最佳，进一步验证了英文单词构词的特点。

（7）注意力头数分析

本节研究了 Iconformer 中不同注意力头的数量对模型性能的影响，实验结果如表 4.13 所示。

表 4.13 不同注意力头数模型性能比较

注意力头数	Accuracy	Precision	Recall	F1-score
2	0.9603	0.9638	0.9674	0.9656
4	0.9613	0.9672	0.9773	0.9722
8	0.9762	0.9724	0.9876	0.9800
16	0.9888	0.9907	0.9808	0.9857
32	**0.9958**	**0.9986**	**0.9916**	**0.9951**

分析可知，随着注意力头数量从 2 逐渐增加到 32，模型性能逐步提升。当注意力头数量为 32 时，模型性能最佳，相较于 2 个注意力头，模型的准确率和精确率分别提高了 3.55%和 3.48%。表明适度增加注意力头数量能够使模型更好地学习输入序列不同部分的关联特征，产生更丰富的特征表示，从而提高模型性能。然而，注意力头数量的增加会显著增加模型的复杂度和参数量，容易导致过拟合。当注意力头数量超过 32 后，模型性能略有下降，说明不能盲目增加注意力头数量，应根据不同模型特点适当设置。

4.4 小结

本章提出一种融合域名解析特征的恶意域名检测方法，获取域名 DNS 解析记录，构建解析特征，以丰富域名的特征表示，并采用基于注意力机制的特征融合模块融合域名文本、注册和解析特征。此外，设计了一种恶意域名检测模型 Iconformer，使模型更充分地学习域名序列的潜在特征表示。其中，MHPSSA 模块通过稀疏性度量关注重要的注意力点积，有效降低模型计算复杂度和参数量，结合卷积模块蒸馏操作，减小模型参数量，从而提高模型泛化性。

在 Cisco Umbrella 合法域名数据集和 CICBellDNS2021 等九个恶意域名数据集上进行了对比、泛化性和消融实验，实验结果表明，域名解析特征能够进一步提高恶意域名检测效果和泛化性，相较于主流恶意域名检测模型，Iconformer 参数量较少，且效果最佳。

总结与展望

本文旨在探索如何有效地丰富并捕获不同类型域名特征，首先，采用工具获取域名 WHOIS 注册信息和 DNS 解析记录两类数据；然后，采用 Transformer 和 CNN 等深度神经网络等技术手段，有效提取域名潜在特征表示；最后，通过在公开数据集上详细实验，结果说明本文所提出的方法能够明显提升恶意域名检测的效果和泛化性。

论文主要研究工作以及成果总结如下：

（1）首先阐述了恶意域名检测任务的研究背景及意义，以确立开展该研究的必要性；其次，分析了国内外在恶意域名检测领域已经取得的研究成果和现存的问题，从而明确目前该领域亟待解决的问题，为后续研究工作提供合理的研究思路；然后，本文详细介绍了恶意域名检测研究中运用到的理论知识和技术手段，如域名、词嵌入、深度神经网络、特征融合方法和模型评价标准等，奠定了后续研究所需的理论基础。

（2）鉴于当前恶意域名检测方法存在的问题，主要集中在对域名特征的多样性考虑不足、域名潜在特征捕获不充分以及泛化性不足。提出了一种融合域名文本和注册特征的恶意域名检测方法。首先，从公开渠道收集包括合法和恶意的域名数据集，获取域名 WHOIS 注册信息，构建域名注册特征，以丰富域名特征表示，获取过程中，忽略部分影响较小注册信息；此外，融合 Transformer 模型和 CNN 模型所提取的域名文本和注册信息的全局和局部特征，以捕获域名潜在特征；最后，在新冠疫情相关数据集上对比实验，结果验证了结合 Transformer 和 CNN 模型效果较优且检测效率较高。同时，在常用基准数据集 CICBellDNS2021 上也获得了较好的效果，证明部分较关键的 WHOIS 注册信息能够有效提升模型泛化能力。

（3）在第三章融合域名文本和注册特征的基础上，获取域名 DNS 解析记录，构建域名解析特征，并采用基于注意力机制的特征融合模块将其与域名文本、注册特征融合，以丰富域名特征表示。此外，在第三章 Transformer 结合 CNN 模型的基础上，设计了一种恶意域名检测模型 Iconformer，结合注意力机制的全局建模和 CNN 的局部建模优点，并引入马卡龙式前馈模块提取多级特征，使模型更充分地学习域名的潜在特征表示。其中，MHPSSA 模块通过稀疏性度量选取重要的注意力点积，有效降低模型计算复杂度和参数量，此外，卷积模块结合蒸馏操作，以突出重要注意力权重，减少冗余信息。将 MHPSSA 模块与卷积模块结合，从而有效减小模型参数量，提高模型泛化性。

在 Cisco Umbrella 合法域名数据集和 CICBellDNS2021 等九个恶意域名数据集上进行了实验，相较于主流恶意域名检测模型，Iconformer 的参数量较少且在四项评价指标均有较大提升。此外，消融实验结果说明卷积模块和 MHPSSA 模块对于模型效果影响较为显著。最后，在不同类型的域名数据集上进行了实验，结果显示，Iconformer 模型在不同类型的域名上均取得了良好的效果，验证了域名注册特征和解析特征模型能够有效提升模型泛化能力，进一步说明 Iconformer 具有较高的实用价值。

尽管本文所提出的融合多类型特征的恶意域名检测方法取得了一定进展，然而，随着网络犯罪分子手段的不断演变，恶意域名检测在网络安全领域仍然是一个亟待解决的挑战，存在许多需要进一步探索和改进的问题：

（1）首先，随着网络犯罪分子生成恶意域名的方法不断演变，恶意域名检测模型的鲁棒性成为亟待解决的问题。未来的研究可以引入对抗样本防御方法，探索对抗性训练、对抗性特征增强和对抗样本生成等技术，以增强恶意域名检测方法的鲁棒性。

（2）其次，本文考虑了域名文本、注册和解析特征，但是未来的研究可以探索更多类型的域名特征，以提高检测模型的准确性和泛化性，并深入研究影响恶意域名检测性能的关键特征。此外，考虑到域名的 DNS 解析记录之间存在关联，未来的研究可以探索利用域名之间的关联关系构建图结构，并采用图神经网络等方法来检测恶意域名。

（3）最后，大多数现有方法为了提升性能而叠加了较深、较复杂的网络，导致模型不易扩展且效率较低。然而，恶意域名检测对实时性要求较高。因此，在未来的研究中，应考虑设计轻量化模型、采用并行计算和分布式训练等技术，以满足实时检测的需求。

参考文献

[1] CNCERT 网络安全信息与动态周报[R]. 国家互联网应急中心, 2024: 1-5.

[2] Zhao H, Chen Z, Yan R. Malicious domain names detection algorithm based on statistical features of URLs[C]//Proceedings of the 2022 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD). IEEE, 2022: 11-16.

[3] Shi Y, Chen G, Li J. Malicious Domain Name Detection Based on Extreme Machine Learning[J]. Neural Processing Letters, 2018, 48(3): 1347-1357.

[4] Antonakakis M, Perdisci R, Nadji Y, et al. From Throw-Away traffic to bots: Detecting the rise of DGA-Based malware[C]//Proceedings of the 21st USENIX Security Symposium (USENIX Security). USENIX, 2012: 491-506.

[5] Ma J, Saul L K, Savage S, et al. Identifying suspicious URLs: an application of large-scale online learning[C]//Proceedings of the 26th Annual International Conference on Machine Learning (ICML). ACM, 2009: 681-688.

[6] Ispahany J, Islam R. Detecting malicious COVID-19 URLs using machine learning techniques[C]//Proceedings of the 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops). IEEE, 2021: 718-723.

[7] Atrees M, Ahmad A, Alghanim F. Enhancing Detection of Malicious URLs Using Boosting and Lexical Features.[J]. Intelligent Automation & Soft Computing, 2022, 31(3): 1405-1422.

[8] Chiong R, Wang Z, Fan Z, et al. A fuzzy-based ensemble model for improving malicious web domain identification[J]. Expert Systems with Applications, 2022, 204(11): 7243-7253.

[9] Wang H, Tang Z, Li H, et al. DDOFM: Dynamic malicious domain detection method based on feature mining[J]. Computers & Security, 2023, 130(10): 3260-3282.

[10] 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.

[11] Pritom M M A, Schweitzer K M, Bateman R M, et al. Data-driven characterization and detection of covid-19 themed malicious

websites[C]//Proceedings of the 2020 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, 2020: 1-6.

[12] 臧小东, 龚俭, 胡晓艳. 基于 AGD 的恶意域名检测[J]. 通信学报, 2018, 39(7): 15-25.

[13] 于光喜, 张棪, 崔华俊, 等. 基于机器学习的僵尸网络 DGA 域名检测系统设计与实现[J]. 信息安全学报, 2020, 5(3): 35-47.

[14] 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.

[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(1): 2401-2413.

[16] Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification[C]//Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS). 2015: 649-657.

[17] Saxe J, Berlin K. eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys[J]. arXiv preprint arXiv:1702.08568, 2017.

[18] Chen Y, Zhou Y, Dong Q, et al. A Malicious URL detection method based on CNN[C]//Proceedings of the 2020 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS). IEEE, 2020: 23-28.

[19] LE H, PHAM H Q, SAHOO D, et al. URLNet: Learning a URL representation with deep learning for malicious URL detection[C]//Proceedings of the ACM Symposium on Principles of Distributed Computing (PODS). ACM, 2017: 1-13.

[20] 罗赟骞, 邬江, 王艳伟, 等. 基于深度学习的集成 DGA 域名检测方法[J]. 信息技术与网络安全, 2018, 37(10): 10-14.

[21] Vinayakumar R, Soman K P, Poornachandran P, et al. DBD: Deep Learning DGA-Based Botnet Detection[J]. Deep Learning Applications for Cyber Security, 2019(1): 127-149.

[22] Peng Y, Tian S, Yu L, et al. A Joint Approach to Detect Malicious URL Based on Attention Mechanism[J]. International Journal of Computational Intelligence and Applications, 2019, 18(3): 1-14.

[23] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J].

Advances in neural information processing systems, 2017, 30(1): 5998-6008.

[24] Maneriker P, Stokes J W, Lazo E G, et al. URLTran: Improving Phishing URL Detection Using Transformers[C]//Proceedings of the 2021 IEEE Military Communications Conference (MILCOM). IEEE, 2021: 197-204.

[25] Zhao K, Guo W, Qin F, et al. D3-SACNN: DGA domain detection with self-Attention convolutional network[J]. IEEE Access, 2021, 10(69): 250-263.

[26] Yang C, Lu T, Yan S, et al. N-trans: parallel detection algorithm for DGA domain names[J]. Future Internet, 2022, 14(7): 209-224.

[27] Liu R, Wang Y, Xu H, et al. Malicious URL Detection via Pretrained Language Model Guided Multi-Level Feature Attention Network[J]. arXiv preprint arXiv:2311.12372, 2023.

[28] Zhang L, Yan Q. Detect malicious websites by building a neural network to capture global and local features of websites[J]. Computers & Security, 2024, 137(10): 3641-3650.

[29] 张震, 张三峰, 杨望. 基于图对比学习的恶意域名检测方法[J]. 软件学报, 10.13328/j.cnki.jos.006964.

[30] 吴涛, 王占海, 张健, 等. 基于 CNN-BiLSTM 迁移自反馈学习的小样本恶意域名检测[J]. 小型微型计算机系统, 2023, 44(3): 602-607.

[31] 余子丞, 凌捷. 基于 Transformer 和多特征融合的 DGA 域名检测方法[J]. 计算机工程与科学, 2023, 45(8): 1416-1423.

[32] 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(10): 2933-2947.

[33] 刘文峰, 张宇, 张宏莉, 等. 域名系统测量研究综述[J]. 软件学报, 2022, 33(1): 211-232.

[34] 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(128): 990-999.

[35] Hahn M. Uniform Resource Locators[J]. EDPACS, 1995, 23(6): 8-13.

[36] 张宾, 张宇, 张伟哲. 递归侧 DNS 安全研究与分析[J]. 软件学报, 2023(01): 1-36.

[37] Zhauniarovich Y, Khalil I, Yu T, et al. A Survey on Malicious Domains Detection through DNS Data Analysis[J]. ACM Computing Surveys, 2018, 51(4): 1-36.

[38] Vinayakumar R, Soman K P, Poornachandran P. Detecting malicious domain names using deep learning approaches at scale[J]. Journal of Intelligent & Fuzzy Systems, 2018, 34(3): 1355-1367.

[39] 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.

[40] Rodríguez P, Bautista M A, Gonzalez J, et al. Beyond one-hot encoding: Lower dimensional target embedding[J]. Image and Vision Computing, 2018, 75(1): 21-31.

[41] Church K W. Word2Vec[J]. Natural Language Engineering, 2017, 23(1): 155-162.

[42] Kenton J D M W C, Toutanova L K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[C]//Proceedings of the NAACL-HLT. 2019: 4171-4186.

[43] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[44] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model.[C]//Proceedings of the Interspeech 2020. Makuhari, 2010: 1045-1048.

[45] Graves A. Long Short-Term Memory[J]. Supervised Sequence Labelling with Recurrent Neural Networks, 2012, 385(1): 37-45.

[46] Pawade D, Sakhapara A, Jain M, et al. Story scrambler-automatic text generation using word level RNN-LSTM[J]. International Journal of Information Technology and Computer Science (IJITCS), 2018, 10(6): 44-53.

[47] 陈冰儿, 劳南新. 基于 LSTM 的许嵩风格歌词生成[J]. 网络安全技术与应用, 2020(8): 49-52.

[48] 钱揖丽, 马雪雯. 基于句子级 LSTM 编码的文本标题生成[J]. 计算机应用与软件, 2021, 38(5): 190-195.

[49] Su C, Huang H, Shi S, et al. Neural machine translation with Gumbel Tree-LSTM based encoder[J]. Journal of Visual Communication and Image Representation, 2020, 71(10): 2811-2822.

[50] Jian L, Xiang H, Le G. Lstm-based attentional embedding for English machine translation[J]. Scientific Programming, 2022, 1(01): 1-8.

[51] 刘婉婉, 苏依拉, 乌尼尔, 等. 基于 LSTM 的蒙汉机器翻译的研究[J]. 计算机

工程与科学, 2018, 40(10): 1890-1896.

[52] 任勉, 甘刚. 基于双向 LSTM 模型的文本情感分类[J]. 计算机工程与设计, 2018, 39(7): 2064-2068.

[53] 於雯, 周武能. 基于 LSTM 的商品评论情感分析[J]. 计算机系统应用, 2018, 27(8): 159-163.

[54] Ma Y, Peng H, Khan T, et al. Sentic LSTM: a Hybrid Network for Targeted Aspect-Based Sentiment Analysis[J]. Cognitive Computation, 2018, 10(4): 639-650.

[55] Zhang W, Chen W, Zhang Z, et al. ITransformer_CNN: A Malicious DNS Detection Method with Flexible Feature Extraction[J]. SSRN, 2022, 2022(1): 1-14.

[56] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[57] 张清, 张文川, 冉兴程. 基于 CNN-BiLSTM 和注意力机制的恶意域名检测[J]. 中国电子科学研究院学报, 2022, 17(9): 848-855.

[58] Kitaev N, Kaiser Ł, Levskaya A. Reformer: The Efficient Transformer[J]. arXiv preprint arXiv:2001.04451, 2020.

[59] Sun Y, Dong L, Huang S, et al. Retentive Network: A Successor to Transformer for Large Language Models[J]. arXiv preprint arXiv:2307.08621, 2023.

[60] 杨成, 芦天亮, 闫尚义, 等. 基于 N-gram 和 Transformer 的 DGA 恶意域名检测[J]. 中国人民公安大学学报（自然科学版）, 2022, 28(3): 100-108.

[61] Xia P, Nabeel M, Khalil I, et al. Identifying and Characterizing COVID-19 Themed Malicious Domain Campaigns[C]//Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy (CODASPY). ACM, 2021: 209-220.

[62] Yuan J, Liu Y, Yu L. A novel approach for malicious URL detection based on the joint model[J]. Security and Communication Networks, 2021, 2021(1): 1-12.

[63] Ren F, Jiang Z, Liu J. A bi-directional lstm model with attention for malicious url detection[C]//Proceedings of the 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE, 2019: 300-305.

[64] Dai Y, Gieseke F, Oehmcke S, et al. Attentional feature fusion[C]//Proceedings of the IEEE/CVF winter conference on applications

of computer vision. IEEE, 2021: 3560-3569.

[65] Hussain M, Cheng C, Xu R, et al. CNN-Fusion: An effective and lightweight phishing detection method based on multi-variant ConvNet[J]. Information Sciences, 2023, 631(1): 328-345.

[66] Gulati A, Qin J, Chiu C C, et al. Conformer: Convolution-augmented Transformer for Speech Recognition[J]. arXiv preprint arXiv:2005.08100, 2020.

[67] Zhou H, Zhang S, Peng J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting[C]//Proceedings of the AAAI conference on artificial intelligence. AAAI, 2021: 11106-11115.

[68] Dai Z, Yang Z, Yang Y, et al. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 2978-2988.

[69] Wang Q, Li B, Xiao T, et al. Learning Deep Transformer Models for Machine Translation[J]. arXiv preprint arXiv:1906.01787, 2019.

[70] Nguyen T Q, Salazar J. Transformers without Tears: Improving the Normalization of Self-Attention[J]. arXiv preprint arXiv:1910.05895, 2019.

[71] Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]//Proceedings of the IEEE/CVF international conference on computer vision. IEEE, 2019: 1314-1324.

[72] Lu Y, Li Z, He D, et al. Understanding and Improving Transformer From a Multi-Particle Dynamic System Point of View[J]. arXiv preprint arXiv:1906.02762, 2019.

[73] Sun X, Liu Z. Domain generation algorithms detection with feature extraction and Domain Center construction[J]. Plos one, 2023, 18(1): 1-25.

[74] Prabakaran M K, Meenakshi Sundaram P, Chandrasekar A D. An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders[J]. IET Information Security, 2023, 17(3): 423-440.

[75] Ding L, Du P, Hou H, et al. Botnet dga domain name classification using transformer network with hybrid embedding[J]. Big Data Research, 2023, 33(1): 100395-100405.

附录 A 相关代码

获取 WHOIS 注册信息代码：

```
import whois
for i in range(len(df['url'])):
    try:
      data = whois.whois(df['url'][i])

      if isinstance(data.get('updated_date', 'None'), list):
         updated_dates = [str(x) for x in data.get('updated_date', 'None')]
         df.loc[i, 'update'] = ','.join(updated_dates)
      else:
         df.loc[i, 'update'] = str(data.get('updated_date', 'None'))

      if isinstance(data.get('creation_date', 'None'), list):
         creation_dates = [str(x) for x in data.get('creation_date', 'None')]
         df.loc[i, 'creation'] = ','.join(creation_dates)
      else:
         df.loc[i, 'creation'] = str(data.get('creation_date', 'None'))

      if isinstance(data.get('expiration_date', 'None'), list):
         expiration_dates = [str(x) for x in data.get('expiration_date', 'None')]
         df.loc[i, 'expiration'] = ','.join(expiration_dates)
      else:
         df.loc[i, 'expiration'] = str(data.get('expiration_date', 'None'))

      df.loc[i, 'registrar'] = str(data.get('registrar', 'None'))
```

获取 DNS 解析记录代码：

```
import dns.resolver
for i in range(len(df['url'])):
   try:
      # Resolve A records
```

```
        answers = dns.resolver.resolve(df['url'][i], 'A')
        a_ip = [answer.to_text() for answer in answers]
        a_types = [answer.rdtype.name for answer in answers]
        a_ttl = [str(answers.rrset.ttl)]

        try:
            # Resolve CNAME records
            cnames = dns.resolver.resolve(df['url'][i], 'CNAME')
            cname_ip = [cname.to_text() for cname in cnames]
            cname_types = [cname.rdtype.name for cname in cnames]
            cname_ttl = [str(cnames.rrset.ttl)]
        except dns.resolver.NoAnswer:
            cname_ip = []
            cname_types = []
            cname_ttl = []

        # Combine A and CNAME results
        combined_ip = ','.join(list(set(cname_ip + a_ip)))
        combined_rdtype = ','.join(cname_types + a_types)
        combined_ttl = ','.join(cname_ttl + a_ttl)

        # Update the DataFrame
        df.loc[i, 'ip'] = combined_ip
        df.loc[i, 'rdtype'] = combined_rdtype
        df.loc[i, 'ttl'] = combined_ttl

    except dns.resolver.NoAnswer:
        df.loc[i, 'ip'] = '解析错误'
        df.loc[i, 'rdtype'] = '解析错误'
        df.loc[i, 'ttl'] = '解析错误'
    except Exception as e:
        df.loc[i, 'ip'] = '解析错误'
        df.loc[i, 'rdtype'] = '解析错误'
        df.loc[i, 'ttl'] = '解析错误'
```