# 第一章 序论

## 1.1 研究背景和意义

网络时代的快速发展带来了一系列的网络安全问题，网络攻击对网民的生活造成的恶劣影响。恶性URL（统一资源定位符）的攻击已经是一种常见的网络攻击手段，如果用户不能分辨链接是否安全，盲目点击之后就会给不法分子可乘之机。各大网站间接为不法分子提供了平台，恶意URL成为了该领域中一个主要媒介。对我们日常生活中用到的计算机设备，一般是通过WEB浏览器搜集信息并且获得在线服务的功能，浏览网页中也会出现更是各样的链接，层出不穷，用户对网络安全的防范意识并不高，很容易进入网络攻击的陷阱造成个人财产损失或者隐私信息泄露，这种危害行为如果不加以制止会危害个人和企业的安全，造成不良的影响。

URL主要作用是用户用来查找相关资源的，但是恶性的URL是对用户产生恶性影响的链接，它会重新定位到相关资源上，可能会出现恶意软件下载或者攻击计算机的行为，正常URL与恶意URL的常见区别如表 1所示，但是恶意URL经常会模仿正常URL的形式，使之难以分辨。随着人们对网络安全的要求逐渐提升，对恶性URL链接的关注也是日益突出。恶性的URL被用于钓鱼网站中骗取用户的个人信息，可以通过共享网络的方式快速传播，也有可能通过大量的垃圾邮件攻击服务器使其瘫痪，严重影响了人们的正常生活。

表 1 正常URL与恶意URL的常见区别

|  |  |
| --- | --- |
| 特征 | 内容 |
| 是否包含IP地址 | 恶意URL 经常使用的方法，使用IP地址代替域名信息 |
| 子域名 | 正常网页URL的子域名一般较短 |
| 域名 | 恶意网页的域名经常使用一些敏感词汇如“account” 、“login” |
| 域名后缀 | 正常网页一般以“.com”等作为结尾 |

URL链接一般是一段长字符串，示意图为图 1，由不同的部分组成，伪造的恶性URL和原始URL是非常相似的，一般不容易被人发现，如果用户点击伪造的URL之后可能会被重定向到钓鱼网站[1]，从而不法分子利用虚假的身份进行诈骗，例如通过伪造网页信息对用户进行诱导填写银行卡号以及密码等操作进行非法访问或者盗取钱财。在一些不良网站上可能会被攻击者植入病毒从而控制设备，窃取机密信息等。

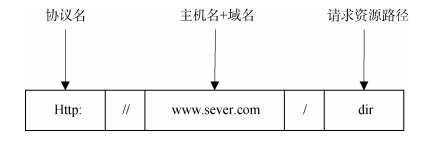


图 1 URL示意图

据研究卡巴斯基报告[2]，在最近几年的时间里，5.45%的个人计算机或者企业服务器都经历过恶意软件的攻击，恶意URL的种类高达上亿，这种恶意的URL攻击需要反病毒的软件进行抵御。

通过反网络钓鱼研究组(APWG)的研究报告[3]可知，网络攻击主要集中在金融行业，造成的经济损失最高达到了每分钟20000美元。CNCERT/CC机构随机抽样到的钓鱼网站大约有251,698个，这些网站通常伪造成我国境内的安全网站诱导用户进行点击访问，大部分的网站是攻击者植入病毒或者垃圾广告造成的，为了诱导消费者获取个人隐私信息从而牟利。攻击的方式非常简单，主要是将URL进行简单的修改便可以欺骗用户，犯罪成本低但是影响非常大。传统的检测方式依赖于规则匹配，难以应对日益复杂的网络环境，所以对于恶意URL的检测研究是刻不容缓的。

## 1.2 国内外研究现状

### 1.2.1 基于黑名单的恶意URL检测

黑名单属于一种静态列表，涵盖了之前被标记为恶性的URL数据，这种静态列表需要专家组进行维护和更新才能即使维护的网络安全。如果访问某个新的URL首先对其进行列表查询，查询已经存在的静态列表中是否已经存在当前URL。相关的网站已经存在，例如PhishTank，该网站会对URL及逆行共享和验证。基于黑名单的检测方式是由Sheng 等人[4]提出的，此种方式对恶意URL的检测准确、高速并且假阳性率较低，但是对于未知的URL是无法进行检测的，目前只能够对已知类型的URL进行检测，存在很大的局限性。对于前身的局限性，Prakash 等人[5]对其进行改进，新增两个组件，可以防止某些恶意URL逃避检测。组件采用了多种试探方式，从静态列表的黑名单中将钓鱼网站进行简单组合对其及逆行扩展，另一个组件对完成组合操作的URL匹配，然后再检测，看结果是否为恶意的URL。

对于黑名单检测方式只能依赖于原始列表的大小，并且更新的时间也是很长的，维护起来耗费大量人工。通常钓鱼网站或者页面的周期很短，一般几个小时就非活跃状态了[6]，所以更新数据库需要非常快速才能保证数据同步，如果数据库来不及完成更新操作会影响后续的检测状态。

### 1.2.2 基于内容检测的恶意URL检测

上述提到的黑名单检测方式完全依赖于原始列表的大小，无法应多复杂多变的网络环境，对于更新和维护也是非常耗时耗力的。根据该项困难，一种根据规则进行匹配的方式应运而生。这种检测方式也是由已知的恶意URL应对未知的，首先对已经存在的恶意URL进行分析，分析出它的网络结构，并且研究一种适用的匹配规则对未知的URL进行匹配检测等。张越等人[10]提出了一种根据URL内容进行检测的方式，这种方式的核心算法为频率逆文档频率算法（TF-IDF）。检测流程首先是选取出现频率较高的五个单词通过搜索引擎进行检索，搜索结果中排名靠前的N个域名和当前网站的域名进行匹配，通过查询匹配的结果判断是否是恶意的URL，匹配规则是如果匹配不上代表是恶意的URL，能够匹配上证明属于合法的网站。经过这种方式的检测，对于恶意URL的检测准确率能够是实现到98%，检测错误率降低到了6%，假阳性率能够降低到1%，但是真阳性率也降低到了89%，某些检测恶意网站的工具例如SpyProxy[7]是根据这种思想实现的。一种名为 Cujo 的系统，最先被Liang等人提出[8]，也是能够实现恶意URL的检测功能，同时还能防止恶意软件的下载和攻击，一般用于到web网站中作为代理，对网页进行检测并且能够快速识别到恶意代码即使提醒。

对于浏览器来说，例如IE浏览器用到了一款恶意URL的检测工具，叫做Cantina，这种工具对于钓鱼网站的检测效果非常准确。对于浏览器中搜索出的结果，通过TF-IDF的数值对其进行判断是否属于恶意的URL或者是恶意的网页，这种方式最先是由Zhang 等人[9]提出的。

### 1.2.3 基于机器学习的恶意URL检测

URL具有统计属性，机器学习利用这一特点对其进行分类，分为良性URL和恶性URL，该方式也有一定的泛化能力，对于未知的新URL也能够检测出来。机器学习的方式采用了提取特征的方式进行训练，将网站上不同信息进行提取，进行检测之前先进行预处理的工作，首先对URL或者HTML（超文本标记语言）进行特征提取的工作然后输入到对应的模型中进行训练。

数据集中特征不相同的分门别类进行存放，在每一类的数据集上对恶性的URL进行检测，这种方式最先是由Vanhoenshoven 等人[27]等人提出的。

在机器学习中使用随机森林算法进行分类检测的实验结果表明，检测的准确性最高。光翔等人[28]对CANTINA进行扩展[10]，通过添加机器学习模块对其目前的功能进行改善，名字也发生变化，改称CANTINA+，解决了假阳性率较高的现象。但是CANTINA+也存在一定的局限性，如果某个钓鱼网站是由图像构成的检测起来存在很大的困难，只能对文本进行分析。CANTINA+的出现虽然准确率提升了92%，但是假阳性率很高的问题依旧存在。一种专门用于检测恶意URL的系统最早是由Gawale 等人[32]设计出来的，能够识别URL或者类似文本中的特征进行检测。

针对URL中存在的 n-gram 特征，根据机器学习算法能够对原始的URL及逆行分类，这种思想是Daeef 等人[33]提出的。Basnet 和 Sung[34]提出的通过机器学习算法进行检测的方式产生了一定的影响力。URL的语法或者与其相关的单词字符等内容可能会被添加到电子邮件中，形成一种恶意的URL，阿泽斯等人[35]采用了朴素贝叶斯算法进行检测。Patil 和 Patil[36]对决策树的多种学习算法进行评估，同时研究了J48决策树、随机森林（RF）。

在平衡数据集上，通过Random Tree、简单购物车和REPTree等算法，能够对恶意URL进行检测。Mustafa AYDIN 和 Kemal BICAKCI 等人[38] 在进行恶意URL检测的过程中应用了“增益比率”和朴素贝叶斯分类器方式。EMUD模型即增强的机器学习模型用于网络钓鱼中对URL进行分类，这种方式最先是由Shweta Sankhwar1 和 Dhirendra Pandey 等[41]人提出的。

Joshi A 等[42]人对长串的URL包含的每一个字符都进行特征的提取然后集成分类之后再开始检测。结合ANOVA(方差分析测试)和XGBoost（极限梯度增强）两种算法提取最显著的特征，然后用数据集分类器XGBoost检测恶意的URL，这种方式最先的提出者是Chen Y C 等人[43]。

Yan X 等人[44]创建了另外一种模型，叫做无监督 UE 模型，对数据进行采样的时候采用了自适应大小的窗口，能够根据输入的文本大小实时调节采样窗口的大小，采用的嵌入编码的方式，优点是能够提升模型的检测精准度，但是URL文本中全部的原始特征是不能完全及逆行还原的。Mamun[45]等人提出的随机森林的检测算法的准确率只够达到97%，对于phishing数据集的检测的准确度只能达到92%，基于检测的精确度不够理想，该模型的使用场景非常有限。

### 1.2.4 基于深度学习的恶性URL检测

深度学习中文本分类的思想值得借鉴[11,12]，深度学习的模型[13]能够自动提取URL中的有有效信息。Bahnsen 等人[14]提出的循环神经网络(Recurrent Neural Network, RNN)[15]检测算法获得了97.6%的准确率。Yang 等人[16]提议对深度学习的模型和n-gram 模型进行结合用于检测。Shibahara 等人[17]利用URL中各个字符之间的语义对其检测。

Woodbridge 等人[18]提出的LSTM[19]即长短期记忆网络的检测算法，这种检测方式的流程首先是将URL进行上下文的特征提取工作，再将其映射成为二维数组，这种方式对于特征提取是不需要人手工进行操作，自动对特征进行提取，神经网络对特征更进一步的学习之后对其进行预测，潘司晨等人[20]提出了此基于卷积神经的模型是。Das 等人[21] 将 CNN 和 LSTM 两种模型进行结合，相较于单一的模型，混合模型的准确率得到提升。对URL进行多分类的半监督模型的提出可以将不同URL的结构和特征分别提取出来，这种模型的提出者Saxe等人[22]，Schiavoni等人[23]提取重定向攻击的 URL域名特征和瞬时特征，将网络优化之后可以得到更高的检测精度。一种多模态的识别算法是赵翠平[24]最早提出的，主要用来检测不良网站。基于上下文信息的检测算法最早是吴海滨等人[25]提出的，这种检测方式采用了特殊的分隔符对URL进行分割，然后通过Word2vec对分隔开的每个字符生成对应的向量空间，通过卷积神经网络将对应的特征提取出来之后进入检测环节。Le 等人[26]将上述的检测方式进行了扩展，填充了一种卷积的分支，用来对字符级别的内容进行特征提取。

连接字符是通过随机产生的，这也是深度学习检测中的一种缺陷，很难对每个字符找到合适的向量空间，况且恶意的链接和正常链接的长相极其相似，难以辨别，所以对其进行特征提取的难度非常大。

## 1.3 本文主要研究内容

对于目前检测技术中存在的难以识别URL上下文关系、提取特征存在不全面以及样本不平衡导致检测准确率低、具体类别的恶意URL检测准确率较低等问题，本文结合Bert与图卷积神经网络测优势进行了以下研究，具体工作内容如下：

1．阐述了恶意URL对网络构成的危害以及正在面临的URL攻击现状，对国内外目前的研究成功实践进行分析总结，并且对已经存在的检测技术阐述了其中的优缺点并且对目前的研究现状进行梳理。

2．提出了基于Bert-FT（Bert·Based·Fine-Tuning Model）的恶意URL分类模型，其使用Bert作为URL字符串的特征提取骨干，并进行下游分类任务微调，相比于基线模型，实现了更优的恶意URL分类检测效果。

3．考虑了URL之间的引用关系，基于引用关系构建了引用网络视图。利用图卷积网络GCN进行图卷积，实现不同URL节点之间的表示传递和聚合，最后利用节点的表征进行节点分类任务。试验结果表明，相比于单独的URL字符串表征，基于图卷积神经网络GCN的恶意URL分类模型能够取得更好的分类检测效果。

4．进一步考虑到URL之间存在不同的视图，以及现有图卷积存在过度平滑的问题，提出了“基于混合视图与自适应门控图卷积的恶意URL检测方法”。首先提出了一种新的卷积方式门控图卷积 (Gate GCN)，Gate GCN首先利用图卷积获取不同卷积层的节点表示，然后利用门控机制融合不同卷积层的结果作为最终的节点表示，能够有效避免过度平滑的问题和保留不同抽象层次的特征，显著增加了模型的鲁棒性。同时消融试验结果表明，所提模型与所添加的模块能够显著提升模型在各种URL任务上的分类性能。

## 1.4 论文组织结构

本文围绕恶意URL检测课题进行研究，论述了研究背景和当前现状，提出了三种新的检测模型。每个章节的内容如下：

第一章为绪论。首先，阐述并分析了本课题的研究背景、意义和现状。介绍了恶意网址对网络和网民的严重后果，介绍了当前国内外研究现状和成果，包括传统的黑名单检测技术和近年来基于深度学习和机器学习的流行检测方法。

第二章对论文所使用的各种技术原理进行了介绍，以及本文所提模型是如何在已有模型基础上进行改进的。主要介绍了Bert预训练大模型的工作原理，以及Bert如何与恶意URL识别任务相结合。

第三章提出了一种基于Bert-FT (Bert·based·Fine-Tuning) 的恶意URL分类模型，首先介绍了Bert模型的结构和原理和各种分类评价指标，然后使用Bert作为URL字符串的特征提取骨干，并进行分类任务微调。并设置了模型对比试验，并对实验结果进行分析，验证了该模型的检测性能。

第四章进一步考虑了URL之间的引用关系，首先基于上述关系构造了对应的视图网络，URL作为节点、URL之间的引用关系构建了一个有向图。其次利用第二章的Bert模型生成URL节点的表征，然后使用图卷积进行消息传递和聚合，综合不同URL节点的表示，并进行最终的节点分类任务。设置了各种消融试验来探究不同条件对最终结果的影响。

第五章再进一步考虑URL之间的多种视图网络，提出了“多视图混合和自适应门控图卷积网络恶意URL检测模型”。首先利用URL间不同的关系构建了多种视图网络，并提出了一种改进的图卷积方式门控图卷积。增加了使用自适应权重模块自动学习边之间的权重关系。最后融合不同视图的表征得到URL节点的表示并取得了最优的恶意URL分类效果。

第六章是总结与展望。总结本文的工作内容，并对研究课题进行展望。

# 第二章 恶意URL检测相关方法技术原理介绍

## 2.1 注意力机制

Bert的核心是基于自注意力机制的Transformer模型，下面首先介绍自注意力机制，一种高效的特征融合方式。

注意力机制能够在分散的数据中将主要特征快速的提取出来，根据任务大小分配不同的资源，能够有效避免信息超载的问题，Bahdanau 等人[63]提出用于优化网络结构。将注意力机制引入到恶意URL的检测模型中的优势在于能够降低对无关特征的关注度。对于输入的信息注意力机制会关注更为重要的信息，基于此特性，当前模型中信息过载的问题能得到有效缓解。

注意力机制的模型如图 2所示。首先需要计算出该模型的权重系数，通过对应函数将 和 联系起来并且将结果进行归一化，最后然对加权求和。首先需要将 和 通过某种方式联系到一起，可采用向量点积乘法的方式如公式(2-1)。

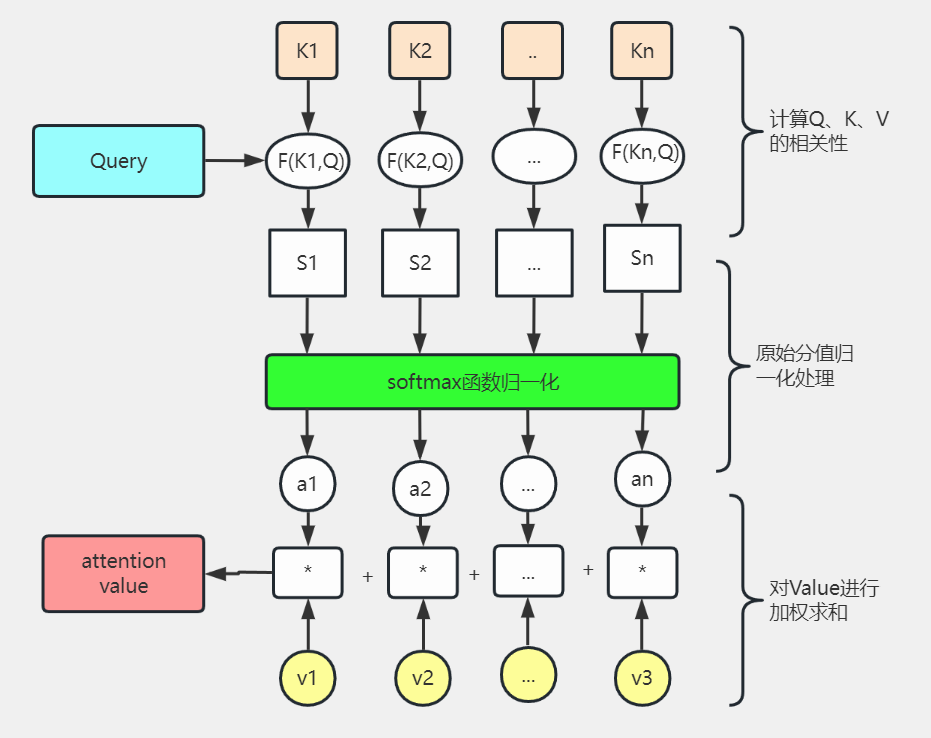


图 2 自注意力机制原理图

(2-1)

通过点积操作计算出的结果需要进行归一化的操作，将其权重的总数为1，需要引入方法，将模型中重要元素的权值显示出来，公式（2-2）所示，计算结果表示的权重，

(2-2)

权重系数加权求和得到注意力机制的结果，计算公式如（2-3）。

(2-3)

## 2.2 Transformer编码器

Transformer的编码器基于注意力机制，架构如图3所示。其负责将输入的URL序列转换为隐藏表示。

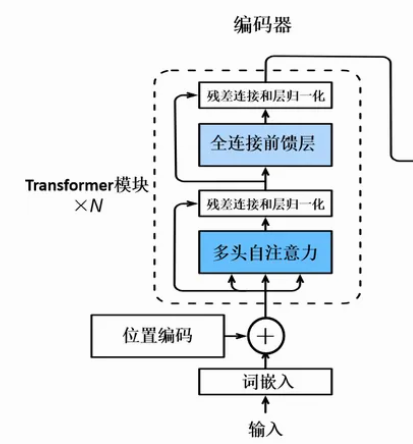


图 3 Transformer编码器架构

Transformer的编码器由堆叠在一起的多个相同层组成，每层包含两个子层：多头自注意力机制和前馈神经网络。在多头自注意力机构中，为输入序列中的每个位置计算加权和，以捕捉输入序列中不同位置之间的依赖关系。前馈神经网络负责对每个位置的隐藏表示进行非线性变换。在真实世界图形数据处理结束时，编码器的输出将被传递给解码器以生成输出序列。

## 2.3 双向记忆网络Bert

Bert（Bidirectional Encoder Representations from Transformers）使用了双向Transformer编码器结构，能够同时学习文本中的前后文信息，这使得它在理解文本中的双向关系上十分强大，可以同时考虑文本中的所有单词和字符。Bert模仅使用Transformer架构的Encoder部分，并在各种NLP任务（文本生成、情感分析等）中取得了良好的效果。这种预先训练的模型在用于下游任务时只需要调整参数即可获得优异的结果，其结构如图 4所示。

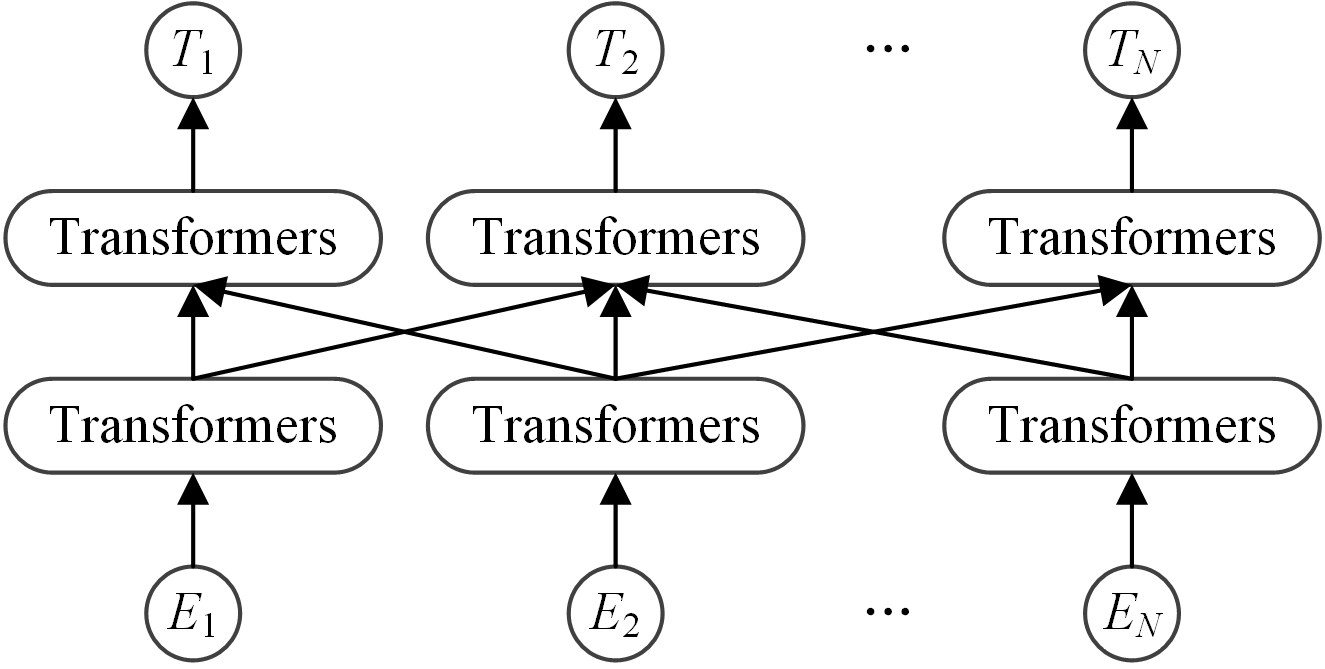


图 4 Bert模型结构

考虑到Bert强大的特征提取能力，本研究使用Bert用于URL字符串特征的生成，并用于下游任务。

### 2.3.1 Bert的资料库与词向量

词嵌入的目的时把单词变成向量进行展示，如果采用One-hot编码的形式无法表现出单词之间的语义关系，对于每一个单词或者字符都可以将其转换成向量，但是向量维度只有某一个方向上才是1，剩余的方向全部是0。

在Bert中使用了Word2Vec[48]该算法，算法对计算的要求也不是很高，可以通过这种方式适应特定的领域，完成特定的任务。近年来多数的NLP都是基于词嵌入的方式进行研究，它能够将URL中的每个字符都进行向量化。CBOW 模型和Skip-gram 模型是Word2Vec算法的具体实现，两者的求解过程分别如公式（2-4）和公式（2-5）所示，图 5展示将字符嵌入到二维空间的示意图，这样每个字符都存在一个向量表征，然后使用这些向量表征进行URL检测。

(2-4)

(2-5)

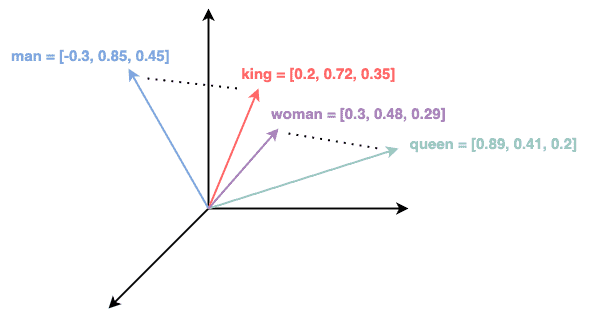


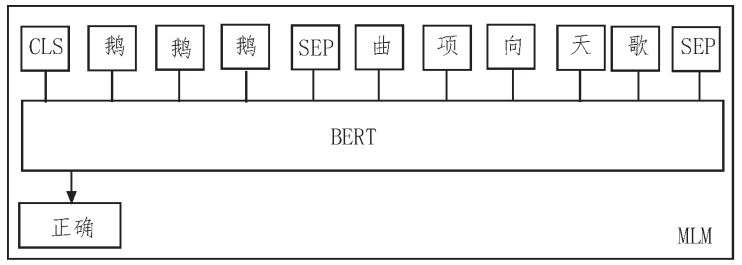
图 5 词嵌入空间

Bert的资料库很大，其将所有资料库进行清洗、去除停用词，分词之后，将每个分词赋予一个索引，并进行训练，因此在输入时只需要输入索引序列，模型内部会自动将索引序列转换为词向量，默认的维度为768维度，代表着768维度的向量空间。

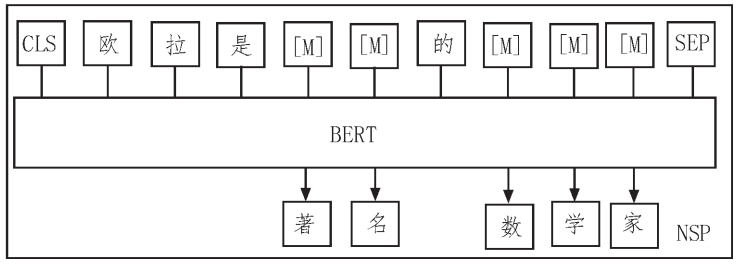
### 2.3.2 Bert的训练流程

Bert在预训练阶段利用下面几种训练方式对语料库进行全面的训练，这能从多种角度学习语料库中的多种高阶语义。

**预训练阶段**：Bert通过预测数百万句子缺失单词进行训练，这种练习有助于Bert建立对语言模式和关系的牢固理解。



**掩蔽语言模型（Masked Language Model，MLM）目标**：，Bert被给定带有一些单词掩码（隐藏）的句子。然后，它基于周围上下文预测这些掩码单词。通过猜测缺失的单词，Bert学会了单词之间的关系，实现了其上下文的卓越性。



**下一句预测（Next Sentence Prediction，NSP）目标**：在NSP目标中，Bert被训练来预测一个句子是否跟随另一个句子。这有助于Bert理解句子之间的逻辑关系，使其能够理解更长的文本。

### 2.3.3 Bert的推理过程

同时在推理阶段，需要对输入到模型当中的字符序列数据进行编码，由于Bert自带语料库与词表，因此需要依据词表获取输入字符序列的索引，形成索引序列，然后经过Bert生成序列的最终表示。具体流程如下：

**词嵌入与上下文词嵌入**将词嵌入看作单词的代码词。Bert通过上下文词嵌入更进一步。与其为每个单词只有一个代码词不同，Bert根据单词在句子中的上下文创建不同的嵌入。这样，每个单词的表示更加微妙，并且受到周围单词的影响。

**Word Piece分词：处理复杂词汇**Bert的词汇就像是由称为子词的较小部分组成的拼图。它使用Word Piece分词将单词分解为这些子词。这对于处理长单词和复杂单词以及处理它以前没有见过的单词特别有用。

**位置编码：导航句子结构**由于Bert以双向方式阅读单词，它需要知道句子中每个单词的位置。位置编码被添加到嵌入中，使Bert具有这种空间感知。这样，Bert不仅知道单词的含义，还知道它们在句子中的位置。

经过上面的步骤，输入一个字符序列数据，Bert就能够输出字符序列数据的768维度的表示，然后作为下游分类任务的输入。

## 2.4 多视图网络

**视图网络（View Network）**指的是在图神经网络中，针对物理实体的不同视图或子图进行学习的一种网络结构。在图数据中，节点（nodes）和边（edges）可以构成不同的视图，每个视图可能包含不同的信息和结构特征。视图网络通过在多个视图上学习，能够捕获到图的多维度特征，从而提高模型对图数据的理解和预测能力。

在恶意URL识别任务中，可以有多个视图，如基于引用关系的视图、基于重定向的视图等，每个视图都可能揭示出不同的URL连接模式和关系。

## 2.5 图卷积神经网络GCN

图卷积神经网络（Graph Convolutional Network，GCN）是一种适用于非欧式空间数据的深度学习方法。它起源于对图形信号处理和卷积神经网络（CNN）的研究，这说明GCN的本质是提取逐层网络图的结构特征。

其次，在原理上GCN和CNN存在相似之处，都可以进行层叠，即多层网络结构可以捕获更复杂的模式和更高层次的特征。在每一层中，节点或像素的特征表示会被转换和组合，以学习数据的更抽象特征。两者的示意图如图 6所示。

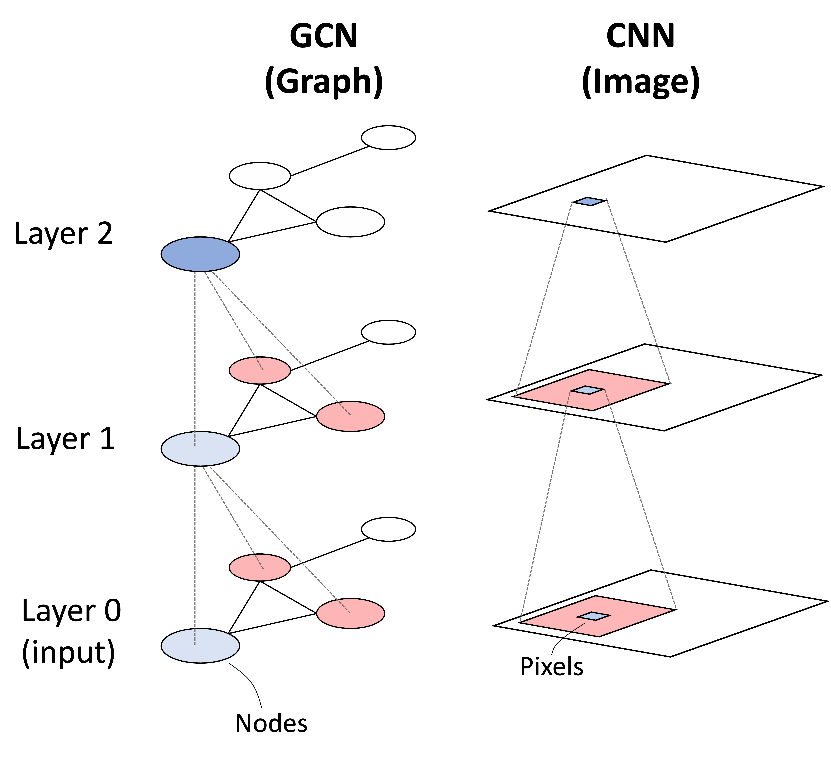


图 6 CNN与GCN原理对比图

GCN的核心思想是，对于图中的每一个节点，我们聚合该节点周围相邻结点的信息（包括自环），然后通过一个聚合函数（如average函数）来整合这些信息，如图 7所示。

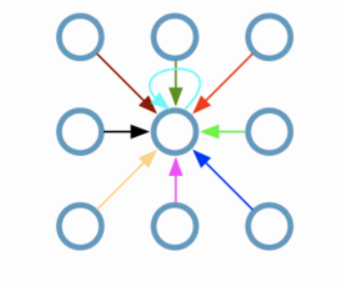


图 7图卷积节点间的信息传递

其具体计算公式如（2-6）。其中，是归一化之后的邻接矩阵，表示带自环的度矩阵，表示节点的特征矩阵，表示神经网络的可训练参数。

 （2-6）

## 2.6 模型微调

模型微调是指进一步训练预训练的模型以适应特定的任务或数据集。这种方法可以提高模型在特定任务上的性能，特别是在数据量有限的情况下。模型微调的一些重要特征如下：

1. 基于预训练的大型模型：通常是指在大型数据集上训练并学习了一些通用特征表示的模型。

2.目的是解决特定任务：微调的目的是让模型能够更好地处理特定的任务，比如特定领域的文本分类等。

3.较小的数据集：微调通常需要一个与目标任务相关的较小数据集。这个数据集可能包含有限的样本，但是它们是针对特定任务的。

4.需要进行模型参数的冻结：在微调过程中，通常会冻结预训练模型的一部分参数，特别是底层的参数，以保持模型学习到的通用特征。然后，只对模型的顶层或新添加的层进行训练，以学习特定任务的特征。

5.较小的学习率：微调通常使用较小的学习率，因为预训练模型已经接近最优解。微调的目的是进行小幅度的调整，而不是从头开始学习。

模型微调的方法主要有以下两种：

1. 基于特征的微调：这种方法涉及冻结预训练模型的参数，仅对任务模块进行训练，以减少需要更新的参数数量，提高训练效率，并保留原始模型的特征提取能力，如图 8左图。

2. 更新所有层的微调：与基于特征的微调相对，这种方法不冻结预训练模型的参数，而是对所有层进行微调，以获得更优的模型性能，图 8右图。

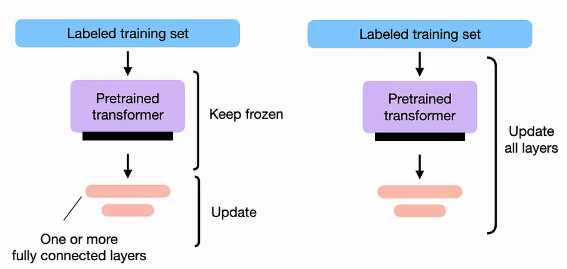


图 8 微调方法

综上所述，模型微调允许我们利用已有的预训练模型，以更高效的方式解决特定问题，特别适用于本文所研究的恶意URL检测任务。

## 2.7 本章小结

本章介绍了本文研究所使用的技术方法原理，以及这些方法原理与本文研究问题的具体相关性。其中最重要的是Bert模型，其是一个预训练大模型，作为的是本文研究的底层模型，用于生成URL字符串的表征，同时对所提模型进行微调，以改善模型生成更优的URL表征。

# 第三章 基于Bert预训练模型的恶意URL检测模型

## 3.1 基于Bert微调模型Bert-FT的恶意URL分类

### 3.1.1 问题定义

基于Bert微调模型Bert-FT的恶意URL分类问题可以定义为：输入URL字符串到模型中，然后输出该URL属于恶意URL的概率，具体的数学表达式如下式（3-1）所示。

 （3-1）

其中，表示输入的字符串序列，，表示批大小，表示URL长度，表示嵌入的维度。表示模型的参数。

### 3.1.2 Bert-FT模型结构

如图 9所示，本文设计的Bert-FT由3层组成：嵌入层负责对语料库进行预训练并生成语料库向量；编码层负责生成输入的嵌入表征；分类输出层对信息层中的信息进行分类以获得意图标签。

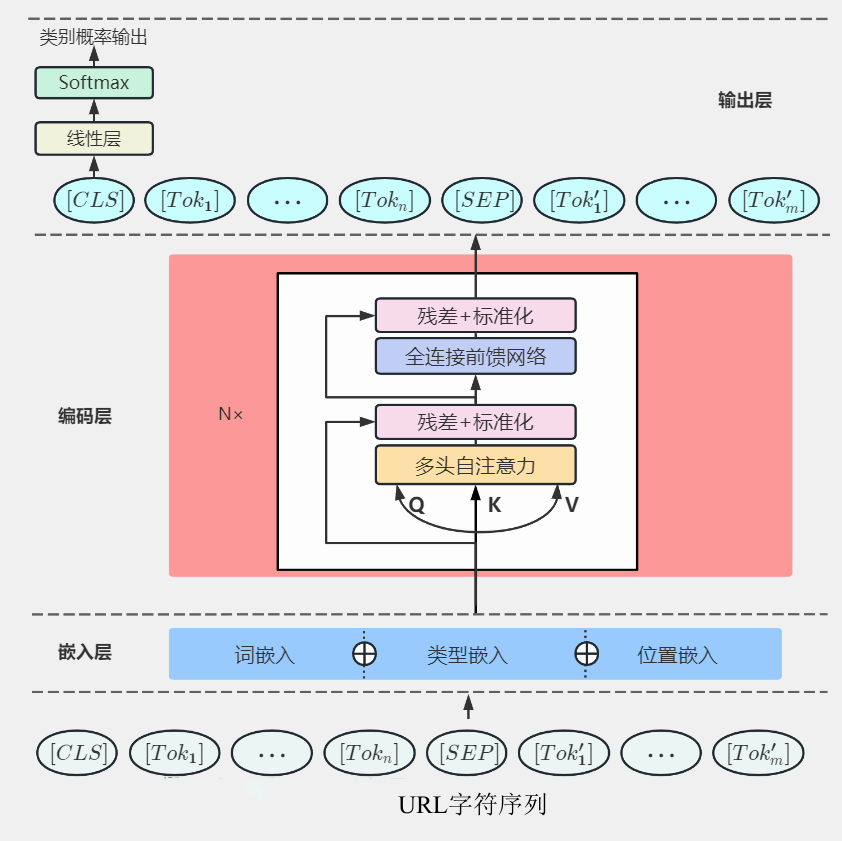


图 9 Bert-FT模型

微调的两种具体方法如下：

1、冻结Bert预训练模型的权重，使其不参与梯度。优化器只更新线性分类层的参数；

2、不冻结Bert预训练模型的权重，使其参与梯度。优化器更新整个网络的所有参数；

## 3.2 数据集与平均指标

### 3.2.1 数据集介绍

本次实验采用了两组数据集，分别取名为ISCX-URL2016 和 Phishing and Benign Websites Dataset (PBWD)。

PBWD数据集通过使用专门的网络爬虫从互联网上收集了大约150万个网页的数据。包含了多个提取的属性以及原始网页内容，包括 JavaScript 代码。数据集支持监督学习和无监督学习，并且已经使用Google Safe Browsing API 为恶意和良性网页添加了类标签。数据集中包含了11个属性，例如网页的 URL、IP 地址、国家地理位置、URL长度、JavaScript代码长度、顶级域名、WHOIS 信息的完整性、HTTPS使用情况、原始网页内容以及网页的分类标签（恶意或良性）。这个数据集适用于深度学习算法，因为它提供了大量的网页数据，并且这些数据可以用于文本分析的非结构化数据输入。

ISCX-URL2016 是加拿大 Dalhousie 大学信息安全卓越中心（Information Security Center of Excellence，ISCX）提供的一个数据集，它用于研究 URL（统一资源定位符）的分类和分析。这个数据集包含了大量的 URL 样本，涵盖了各种不同类型的 URL，例如恶意 URL、垃圾邮件 URL、正常 URL 等，是网络安全研究领域的重要资源。

数据中的部分URL链接的长度低于200，采用空格进行填充操作，长度超过200的进行字符串的截取。在数据集上进行采样，其中的良性链接和恶性链接各占一半，ISCX-URL集合中抽取20000个样本，名称取为ISCX-URL20000，PBWD集合中抽取30000个样本，命名为PBWD30000，抽取到的样本中各含有一半的恶性链接，表 2展示了部分URL字符串的详细信息。

表 2 部分数据样本展示

|  |  |
| --- | --- |
| URL样本 | 标签 |
| ijsbaanapeldoorn.nl/fotos-videos.html?vsig4 0=20 | malicious |
| mail.verboven.com.tr/netbank/logon | malicious |
| tlcllc.com/home/7-ron-potter-blog/85-horns-of-a-dilemma.html | malicious |
| ipl.com/products/p0000.uk.php | benign |
| olavi.com/internet.htm | benign |
| selfabuserecords.net/home.htm | benign |

### 3.2.2 评价指标

判断该模型是否准确可靠，设定了六个参考标准，根据以下的指标对性能进行评估：准确率Acc，精确度Pre，假阳性率FPR，召回率Rec，F1值，在检测的过程中，将模型检测出来的结果和真实结果对比，结果如果显示为真阳性就代表该链接URL确实为恶意的URL，假阳性就表示为良性链接，归为恶意链接是错误的；真阴性代表对应的URL分类未正确的，确实是良性的链接；假阴性代表该链接虽然是良性的，但是分类错误，误入了恶意分类中。根据四种检测结果进行融合，构建一个矩阵进行展示，如表3-5：

表 3 混淆矩阵

|  |  |  |
| --- | --- | --- |
| 真实\检测 | 1 | 0 |
| 1 | TP | FN |
| 0 | FP | TN |

TP的含义代表了属于恶意的URL并且分类结果准确，FN表示的是良性的URL并且分类结果是正确的，FP属于恶意的URL但是分类结果是错误的，FN表示的良性的URL但是分类结果是错误的。

准确率(Acc):检测正确的数量和所有样本的比例，计算公式如下（3-1）。

（3-1）

精确度(Pre):检测为恶意URL中检测结果正确的比例，的计算公式如下（3-3）：

（3-3）

假阳性率(FPR)：所有良性样本中检测错误的比例，计算结果如下（3-4）：

（3-4）

召回率(Rec)：所有恶意样本中检测为恶意并且正确的比例，计算公式（3-5）：

（3-5）

F1值(F-Score)：属于一个综合的指标，结合了精确度和召回率指标，数值越大代表了该模型的鲁棒性越强的，对抗未知风险的性能强，计算公式如下（3-6）：

（3-6）

## 3.3 结果分析与评价

### 3.3.1 训练配置

用于训练模型的机器配置情况如下。

表 4 机器配置表

|  |  |
| --- | --- |
| 配置 | 内容 |
| GPU 处理器 | NVIDIA GeForce GTX1080Ti 11GB |
| 内存 | 32GB |
| 操作系统 | Windows10 |
| 编程语言 | Python Pytorch |

模型的网络层参数与训练参数如下表所示：

表 5 模型参数及训练参数表

|  |  |
| --- | --- |
| 配置 | 内容 |
| 批次大小 | 32 |
| 学习率 | 0.0001 |
| 优化器 | Adam |
| Epoch | 60 |
| 全连接层神经元个数 | 256 |
| 损失函数 | 多元交叉熵 |

### 3.3.2 Bert-FT模型分类结果

Bert-FT模型在该数据集上的进行训练和验证的Loss结果如图 10，由图可知，在经过15轮的训练之后图像渐渐平稳。训练结束之后采用20%的验证集样本测试该模型的检测性能。

|  |
| --- |
|  |
| 图 10 LOSS随训练轮次的变化曲线  在相同数据集上设立对照试验，将传统LSTM、卷积网络Text-CNN和Bert-FT的检测结果进行汇总，如表6所示: |

表6 不同模型的检测性能

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模型 | Acc | Pre | FPR | Rec | F1 |
| LSTM | 94.63% | 95.16% | 3.32% | 92.67% | 93.92% |
| Text-CNN | 95.23% | 96.18% | 2.31% | 94.15% | 95.15% |
| Bert-FT | 96.79% | 96.59% | 2.05% | 94.85% | 95.88% |

可以看出Bert-FT冻结参数模型的分类效果显著高于LSTM和Text-CNN等传统模型，而且误报率在所有模型当中最低，证明了Bert-FT模型即使拥有较少的可训练参数，仍然拥有良好的特征提取能力。

表7 不同微调方式检验分析结果

|  |  |  |
| --- | --- | --- |
| 指标 | Model(平均值+-标准差) | |
| 是否冻结Bert参数 | 是 | 否 |
| Acc | 96.25%±0.04% | 96.79%+0.14% |
| Pre | 96.35%±0.22% | 96.59%+0.39% |
| FPR | 2.43%±0.13% | 2.05%+0.62% |
| Rec | 94.26%±0.13% | 94.85%+0.42% |
| F1 | 95.37%±0.12% | 95.88%+0.36% |

表7显示了Bert-FT模型是否冻结Bert特征提取骨干的参数模型在各项指标上的实验结果对比，由此可知不冻结参数模型的各项指标均优于参数冻结的模型，全参数Bert-FT的检测性能高于非全参数Bert-FT，具有一定的优越性。

## 3.4 本章小结

本章使用Bert预训练大模型作为URL字符串序列的特征提取器，然后在Bert后面添加一层全连接分类层进行二分类任务。并且对是否冻结Bert预训练模型的参数进行了消融试验，结果证明使用较小的学习率对Bert预训练模型进行参数更新能够取得更优的效果。

# 第四章 基于URL引用视图和GCN的恶意URL检测模型

本章考虑了URL之间存在互相引用的关系，为了提取这种关系网络特征以及对恶意URL识别的影响。本章节在Bert-FT模型的基础上，引入URL引用视图，以URL为节点，根据引用关系构建邻接矩阵图网络，随后利用GCN进行URL节点之间的消息传递和信息聚合。

## 4.1 基于URL引用关系的网络构建

URL之间的引用关系通常指的是一个网页或资源通过超链接指向另一个网页或资源的情况。这种关系在互联网上非常普遍，是网页之间相互连接和信息共享的基础。以下是几种常见的URL引用关系：

1.超链接引用：最常见的引用关系，一个网页通过`<a>`标签中的`href`属性链接到另一个网页。

2.资源引用：网页可能会引用其他资源，如图片、CSS样式表、JavaScript脚本等。这些引用通过`<img>`、`<link>`、`<script>`等标签实现。

3. API调用：应用程序接口（API）允许一个应用通过特定的URL请求数据或服务，这也是一种引用关系。

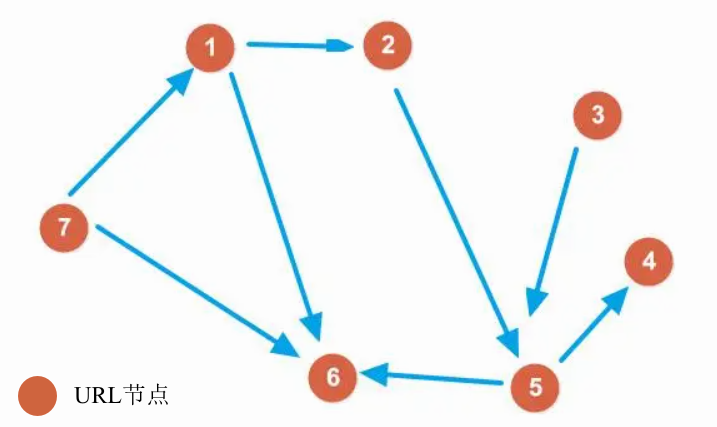


图 11 URL节点间的引用关系

如图 11所示，引用关系不仅帮助用户在不同网页之间导航，也是互联网内容组织和信息传播的重要机制。通过这些引用，信息得以在不同的网站和资源之间流动和共享。重定向关系则涉及到URL的动态行为，即一个URL可能会根据服务器的配置或用户的请求被自动转换为另一个URL。因此，进行建模时考虑URL间这种引用关系是必要的，因为这种引用关系能够提供URL之间拓扑特征，提高模型的鲁棒性。

我们以URL为节点，如果URL之间存在引用关系，我们就建立URL节点之间的有向边，最终得到URL引用关系的整图网络。

## 4.2 基于Bert模型的URL节点特征生成

在图数据结构中，节点特征的生成是一个关键的步骤，它直接影响到图分析和机器学习模型的性能。基于Bert-FT（Bert Fine-Tuning）模型的URL节点特征生成是一种新的思路，它能够深入理解URL的语义信息，并生成高质量的节点嵌入。

前一章已经解释了Bert-FT模型是一种预训练的深度双向表示模型，它通过对大量文本数据进行预训练来学习丰富的语言特征和模式。将Bert-FT应用于URL节点特征生成，可以利用其强大的语义理解能力，捕捉URL中的关键信息和上下文关系。与传统的基于特征工程的方法相比，Bert-FT可以自动学习URL的内在特征，而不需要手动设计特征提取规则，大大减少了特征工程的工作量，提高了特征的表达能力。

在实际应用中，Bert-FT模型可以通过微调（Fine-Tuning）的方式，针对特定的URL数据集进行优化。微调过程允许模型在保持预训练知识的同时，进一步学习特定任务的特定特征。这种方法不仅可以提高模型在特定任务上的表现，还可以确保生成的节点嵌入更加符合实际应用的需求。

Bert-FT模型通过深入理解URL的语义内容，生成的嵌入能够更好地捕捉节点间的复杂关系，从而为接下来的图算法提供更加丰富和精确的信息。

## 4.3 基于Bert和GCN的恶意URL检测模型构建

本小节利用上一章节的Bert-FT模型和GCN图卷积模块，构建Bert-FT和GCN的恶意URL检测模型。具体来说，使用Bert-FT模型生成所有URL节点的嵌入表示，然后使用GCN进行特征传递和融合，最后进行二分类。

此外，我们还将Bert模型嵌入到GCN模块当中，形成Bert-GCN-FT模型，同样对该模型进行消融试验，包括是否冻结Bert预训练模型的参数、GCN的卷积层数对最终分类效果的影响。

### 4.3.1 Bert-GCN模型

Bert-GCN模型是一种深度学习模型，结合了Bert和图卷积网络（GCN）。在Bert-GCN模型中，首先使用预训练的Bert模型为每个URL节点生成嵌入式表示，其中包含节点的语义和上下文信息。

接着，将这些Bert生成的节点嵌入作为输入，传入图卷积网络（GCN）中进行图卷积操作，从而在图结构中传播节点之间的信息。图卷积操作可以有效地捕捉节点之间的关系和结构，从而更好地学习节点的特征表示。

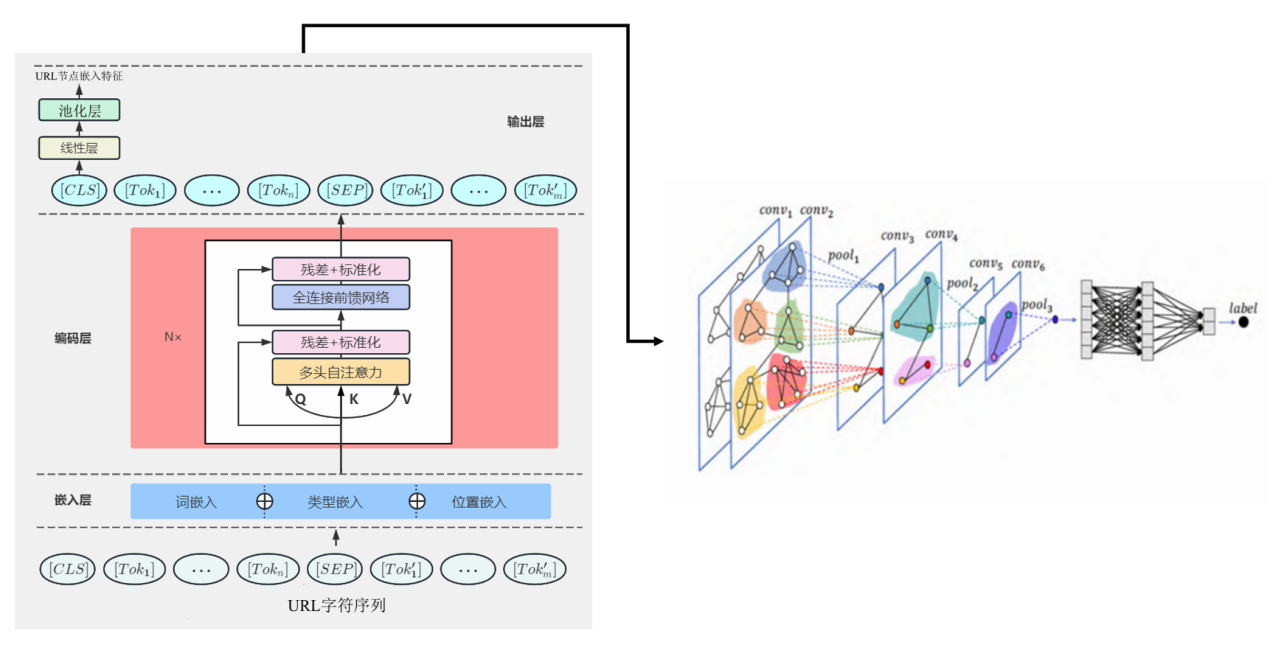


图 12 Bert-GCN模型

最后，将经过图卷积操作后的每一个URL节点表示传入一个分类器，进行二分类任务。在整个训练过程中，Bert模型的参数不参与更新，只有GCN和分类器的参数会被训练。通过结合Bert和GCN，Bert-GCN模型可以充分利用文本信息和图结构信息，从而提高对图数据的表示学习和分类性能。具体结构如图 12所示。

### 4.3.2 Bert-GCN-FT模型

Bert GCN-FT模型是基于Bert GCN模型的改进，其中FT代表微调，即微调。与Bert-GCN模型不同，在Bert-GCN-FT模型中，Bert模型的参数也会被更新，即在训练过程中对Bert模型进行微调。

具体来说，Bert-GCN-FT模型首先使用预训练的Bert模型生成节点（例如URL）的嵌入表示，然后将这些表示传入图卷积网络（GCN）中进行图卷积操作，类似于Bert-GCN模型。不过在Bert-GCN-FT模型中，除了GCN和分类器的参数会被训练外，还会对Bert模型的参数进行微调，以更好地适应特定的图数据任务。

通过在更新中修正Bert模型的参数，Bert-GCN-FT模型可以进一步提高其在图形数据上的表示学习和分类性能，因为Bert模型本身已经在大规模文本数据上进行了预训练，并进行了微调。

## 4.4 模型效果分析与评价

### 4.4.1 数据集介绍

数据集ISCX-URL来自UNB（加拿大网络安全研究所），其中包括四种类型的恶意URL：垃圾邮件、网络钓鱼、恶意软件和污损URL[45]，可以满足本实验的所有要求。训练集和测试集分别占80%和20%。表 8显示了良性URL和四种类型的恶意URL的示例。

表 8 ISCX-URL数据集部分样本

|  |  |
| --- | --- |
| URL类型 | URL样本 |
| phishing | http://gbipengayoman. com/html/dto. php |
| malware | http://dl.helianthemum. com/download/dyn/uninstall/package gamesdesktop DYNPACKAGES.exehttp://ru |
| defacement | mahks. com/index.html |
| span | http://amber.ch.ic.ac.uk/archive/al1/8233. html |
| benign | <http://vnexpress.net/tin-tuc/the-gioi/phap-co-the-bantau-mistral->cho-trung-quoc-3211786.html |

本次实验还验证了来源于Kaggle 社区的Malicious phish 数据集中的URL，该数据集容纳了641119条记录数据，对每个类别的恶意URL都进行了二分类和多分类的实验以此验证该模型的泛化性能。

### 4.4.2 引用视图构建

构建URL参考网络视图的具体操作过程如下。使用URL引用关系构建视图是指在处理网络数据时通过URL之间的引用关系构建网络视图，以便更好地分析和理解网络结构。在这种情况下，URL可以表示网络中的节点，URL之间的引用关系可以表示节点之间的连接或关联。构建视图的过程通常包括以下步骤：

数据收集：首先需要收集网络数据，包括URL及其引用关系。这些数据可以通过网络爬虫、API接口等方式获取。

数据预处理：对收集到的数据进行预处理，包括数据清洗、归一化和解析URL之间的引用关系等。

构建网络视图：根据URL之间的引用关系构建网络视图，可以使用图论中的相关方法来表示网络结构，例如邻接矩阵、节点表示等，具体如图 13的邻接矩阵所示，其中1表示该对应的两个节点存在连接，0则相反。

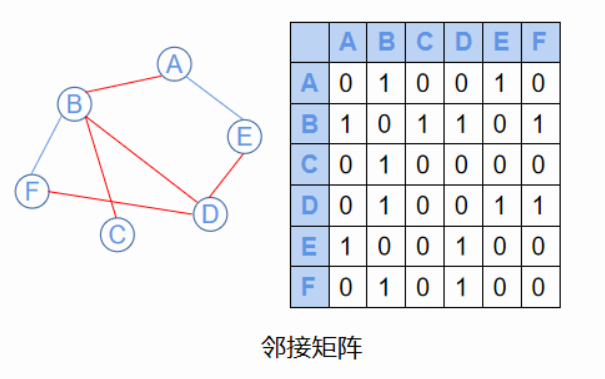


图 13 邻接矩阵示意图

在获取URL引用视图之后，将80%的节点用作训练数据，20%的节点用作测试数据，制作数据集。具体步骤如下：

1. 将获取的URL引用视图中的节点按照80%和20%的比例分割为训练集和测试集。

2. 对训练集，将一部分节点进行MASK操作，即随机选择其中一定比例的节点作为训练的MASK节点。

3. 对测试集，进行相同的MASK操作，作为测试集的MASK节点。

通过这种方式制作数据集，可以有效地利用URL引用视图中的节点信息进行训练和测试，模型能够更好地理解网络结构和节点之间的关系，从而提升模型的性能和泛化能力。

### 4.4.3 Bert-GCN模型分类效果

为了验证Bert-GCN模型的检测性能，本章在两个数据集上进行了二元和多分类实验。来自ISCX-URL和Malicious phish的这两个数据源的样本都参与了模型的检测实验，使实验更具说服力。结果如图图 14、图 15所示，展示了Bert-GCN模型在两个数据集上的二元和多分类结果。

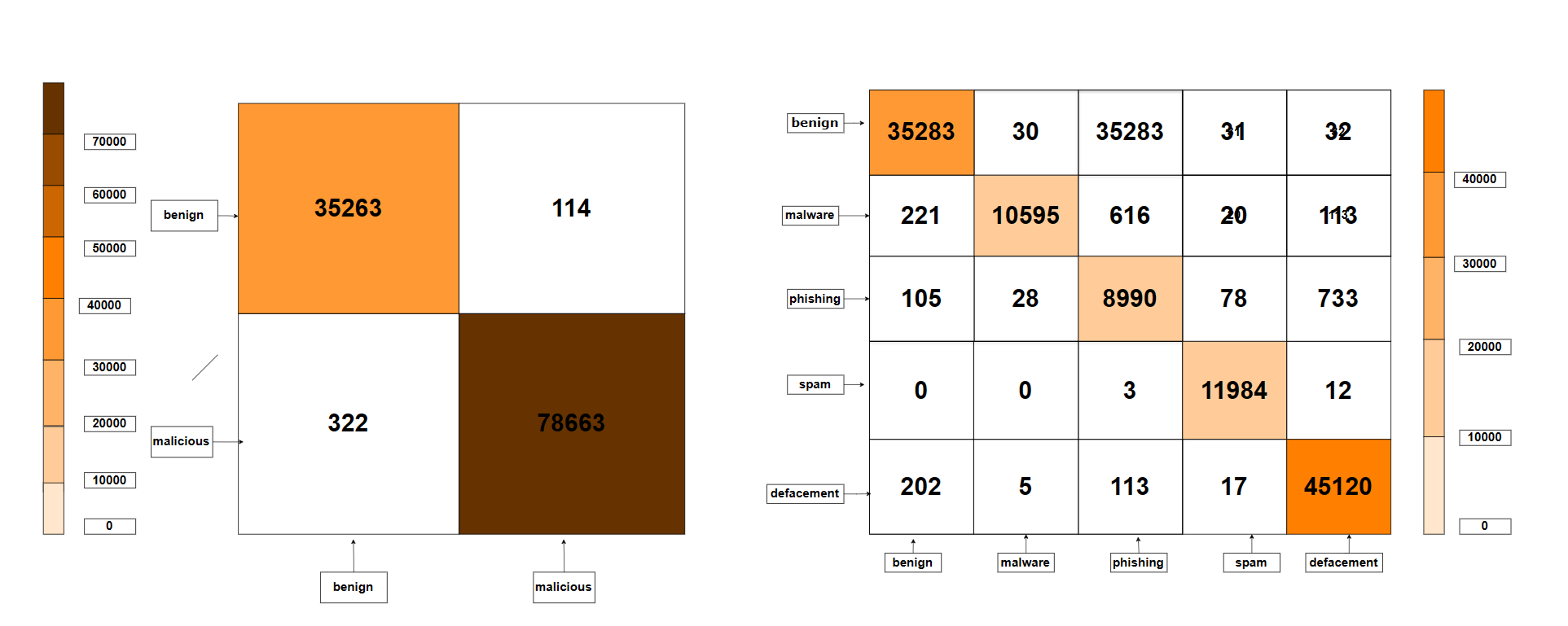


图 14 ISCX-URL数据集上Bert-GCN模型二分类和多分类的实验结果

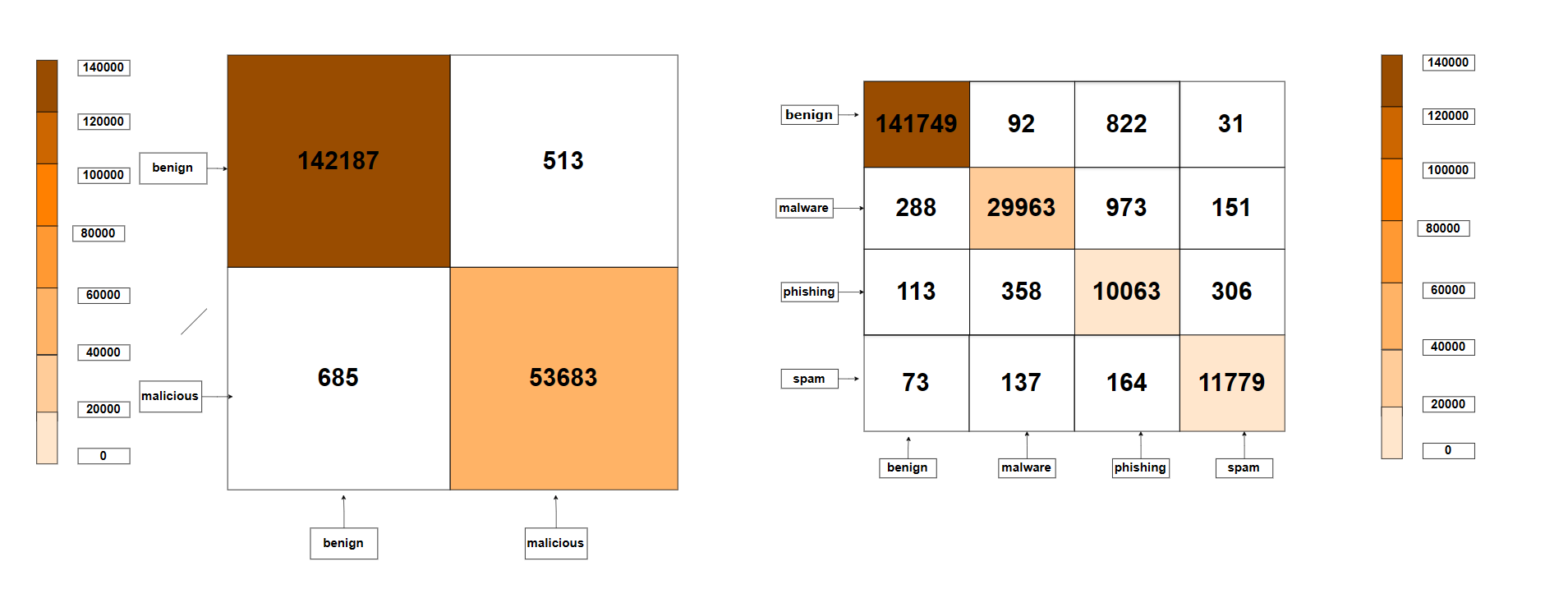


图 15 Malicious phish数据集上Bert-GCN模型二分类和多分类的实验结果

对于不同模型在不同数据集上检测性能各项指标的具体数据结果如下表所示：

表 9 Bert-GCN模型在两种数据集上二分类的实验结果

|  |  |  |  |
| --- | --- | --- | --- |
| 数据集 | precision | recall | F1-score |
| Malicious phish | 97.48% | 97.65% | 97.55% |
| ISCX-URL | 97.89% | 97.87% | 97.86% |

二分类的实验结果如下：Bert-GCN模型在Malicious phish数据集和ISCX-URL都取得了较好的结果，但是在ISCX-UR数据集上的效果明显优于Malicious phish数据集。

表 10 Bert-GCN模型多分类具体类别检测精确率结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模型 | benign | phishing | malware | span | defacement |
| Malicious phish | 97.53% | 97.40% | 97.19% | 97.80% | 97.07% |
| ISCX-URL | 97.40% | 97.54% | 97.42% | 97.75% | 97.48% |

多分类的实验结果如下：相比于二分类，多分类在恶意URL当中进行了更细致的分类。结果展示了在ISCX-UR数据集上的效果明显优于Malicious phish数据集。

表 11不同模型在Malicious phish数据集上的表现

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模型 | benign | phishing | malware | span | defacement |
| Bert-GCN | 97.53% | 97.40% | 97.19% | 97.80% | 97.07% |
| Bert-FT | 97.20% | 96.52% | 95.98% | 96.00% | 95.75 % |
| Text-CNN | 95.62% | 95.58% | 95.55% | 95.08% | 95.11% |
| LSTM | 94.32% | 94.41% | 94.33% | 94.28% | 94.00% |

另外，我们也对比了不同模型在Malicious phish数据集上进行多分类的表现，具体结果如表 11所示，从中可以发现，Bert-GCN模型拥有最好的多分类效果，证明了考虑URL之间引用关系特征能够显著提升模型的鲁棒性。

### 4.4.4 Bert-GCN-FT模型效果

相比于Bert-GCN模型，Bert-GCN-FT模型允许Bert预训练模型的参数进行梯度更新。因此，此小节探究的是Bert预训练的参数是否参与更新对URL分类结果的影响。

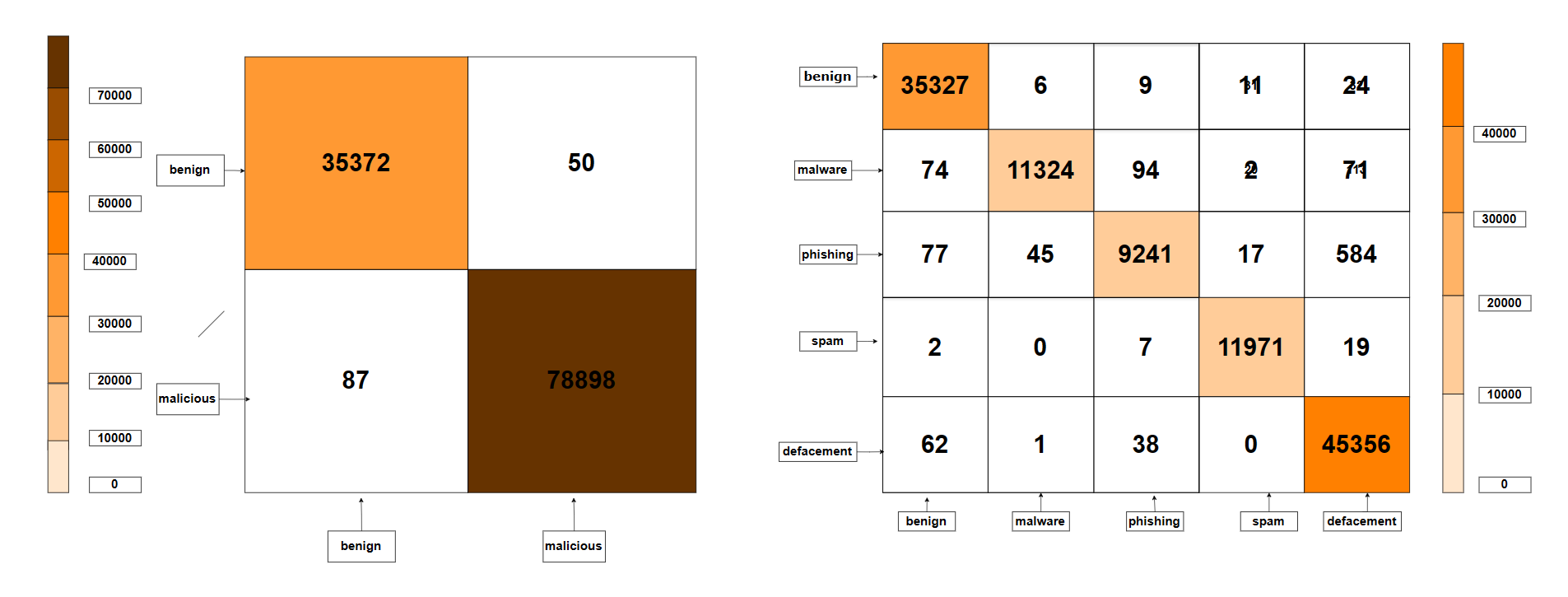


图 16 ISCX-URL数据集上Bert-GCN-FT模型二分类和多分类的实验结果

表 12 Bert-GCN-FT模型在两种数据集上二分类的实验结果

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | precision | recall | F1-score |
| ISCX-URL | 97.60% | 98.01% | 97.80% |

表 13 Bert-GCN-FT模型在两种数据集上多分类的实验结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模型 | benign | phishing | malware | span | defacement |
| ISCX-URL | 98.40% | 98.54% | 98.42% | 98.25% | 98.48% |

由表 12、表 13和图 16可以看出Bert-GCN模型拥有良好的分类性能，在两个数据集上都取得了不错的分类性能。在二分类任务上的F1-score取得了98%的效果，证明其能够较好的区分正类和负类样本。在对正类样本更为细致的多分类任务当中，对于每一种URL类型的二分类也取得了很好的效果，平均准确率都在98%以上。

表 14 ISCX-URL 数据集上Bert-GCN-FT和Bert-FT二分类的实验结果对比

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | precision | recall | F1-score |
| Bert-GCN | 96.89% | 97.87% | 97.51% |
| Bert-GCN-FT | 97.79% | 98.35% | 97.86% |

ISCX-URL 数据集上Bert-GCN-FT和Bert-FT二分类的实验结果对比展示出Bert-GCN-FT相比于Bert-FT能够提升较大的分类精度。

表 15 ISCX-URL 数据集上Bert-GCN-FT和Bert-GCN多分类的实验结果对比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模型 | benign | phishing | malware | span | defacement |
| Bert-GCN | 98.53% | 99.40% | 92.19% | 98.80% | 98.07% |
| Bert-GCN-FT | 99.40% | 99.54% | 98.42% | 99.75% | 98.48% |

ISCX-URL 数据集上Bert-GCN-FT和Bert-GCN多分类的实验结果对比，对每一种恶意URL进行分类，Bert-GCN-FT效果更加。

## 4.5 本章小结

本章节考虑了URL之间的引用关系，构建了URL引用整体视图网络，建立了基于Bert和GCN的恶意URL检测模型。分别研究了Bert预训练模型参数是否参与更新对分类结果的影响。结果表明，考虑URL之间的引用关系，能够显著提升模型的分类性能及其鲁棒性。

# 第五章 基于多视图和自适应门控GCN的URL检测模型

上一章节利用基础的URL引用视图和普通的GCN网络就取得了较好的效果，但是仍然存在一些问题，例如单纯的引用视图无法完全考虑到URL之间的多重关系，简单的GCN网络相较于新颖的特征提取模块特征提取能力较弱。因此，本章节旨在探讨不同URL视图对模型分类结果的影响，以及不同结构的GCN模块对模型的性能影响。

## 5.1 多视图构建

多视图的目的是挖掘URL数据中的多个关系，构建多个有向边缘网络视图，然后将它们添加到模型中。如图 17所示，多视图首先对每个视图执行相同的图卷积操作，以获得该视图下的节点表示，然后通过注意机制或门控机制进行特征融合或最终节点表示。

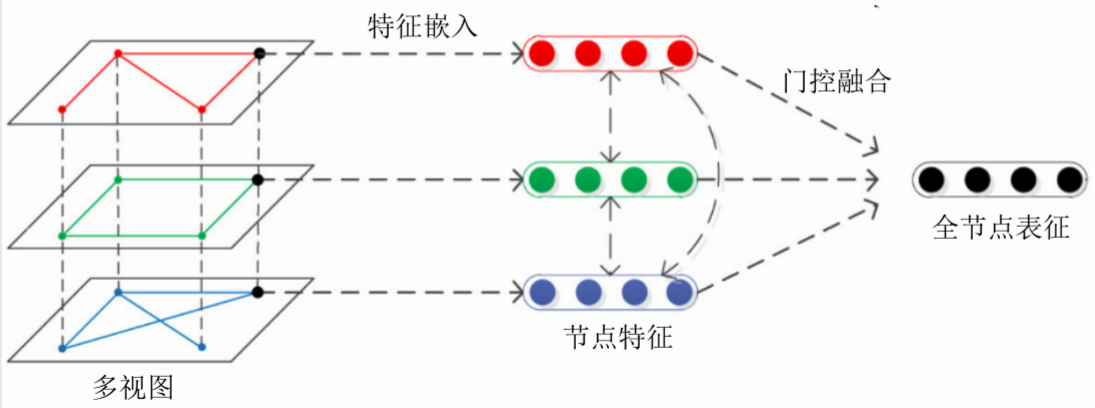


图 17 多视图融合技术路线图

### 5.1.1 引用视图

引用关系不仅帮助用户在不同网页之间导航，也是互联网内容组织和信息传播的重要机制。因为这种引用关系能够提供URL之间拓扑特征，提高模型的鲁棒性。

上一章节以URL为节点，如果URL之间存在引用关系，建立URL节点之间的有向边，最终得到URL引用关系的整图网络。本章节沿用上一章节的引用URL视图。

### 5.1.2 重定向视图

一种常见的网络攻击方式就是URL重定向到恶意网站，如图 18所示，当访问伪装为正常URL时则会被重定向到恶意URL。所以模型将URL间的这种重定向关系构建为一个网络视图，以表征这种恶意的导向关系。在网络中，节点代表URL，边代表URL之间的重定向关系。这种视图的构建对于理解和预测网页重定向行为至关重要。

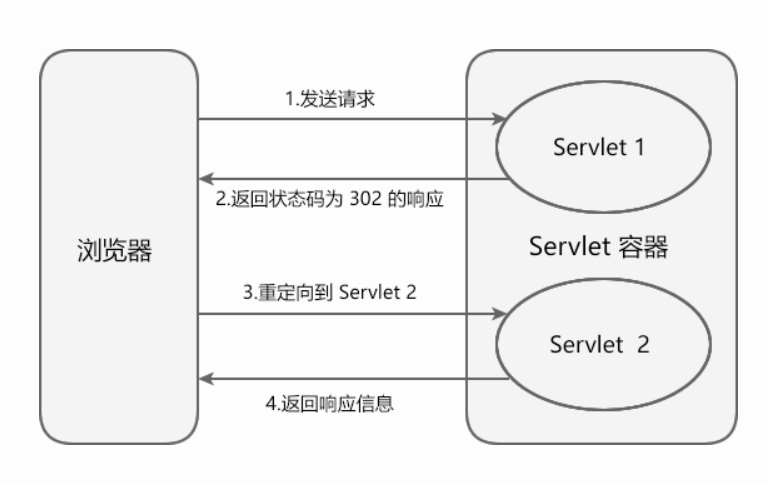


图 18访问URL时的重定向

构建重定向视图的第一步是识别和收集URL之间的重定向链。这通常是通过跟踪HTTP重定向响应来完成的。当用户请求一个URL时，服务器可能会返回一个重定向状态代码，指示客户端应该请求另一个URL。通过分析这些响应，我们可以确定哪些URL节点是相互连接的，即存在重定向关系。

接下来，我们需要根据收集到的重定向数据构建视图网络。这个网络是一个有向图，其中的节点是URL，边是重定向关系。每个节点都与一个或多个节点相连，表示从一个URL到另一个URL的重定向。

在构建视图网络之后，我们需要对网络进行预处理，以便它能够被GCN处理。这包括标准化节点特征、调整边权重以及可能的节点和边的归一化。

### 5.1.3 同源视图

URL同源视图这种方法的核心在于利用节点的同源性来构建视图网络，从而增强模型对节点特征的理解和表达能力，一个站点的URL同源协议如图 19所示，一个域名之下存在很多的URL页面，同源视图就是建立这种关系，因为恶意网站的域名之下通常都是恶意URL。

在构建同源视图网络时，首先需要识别网络中的同源节点。通过识别这些节点，我们可以将它们归类到不同的同源组中。每个同源组可以被视为网络中的一个子图或视图，其中只包含同源节点及其相互之间的连接。

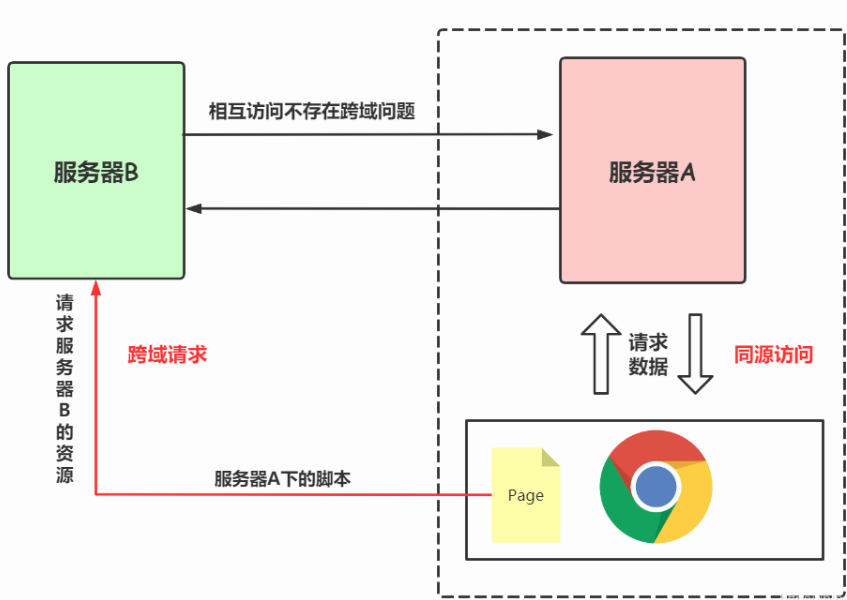


图 19 同源URL示意图

接下来，对于每个同源视图，我们需要构建一个图结构来表示这些节点及其关系。在这个图结构中，而边则表示节点之间的连接关系，如超链接、共享资源等。通过这种方式，每个同源视图都能够捕捉到同源节点之间的局部结构和关系。

在GCN中，节点的特征会通过图卷积操作进行更新。对于同源视图网络，图卷积操作可以专门针对每个视图进行，以利用同源节点之间的相似性。这样，每个视图的节点特征在经过图卷积后，不仅包含了自身信息，还融入了来自同源节点的信息，从而提高了模型对节点特征的表达能力。

## 5.2 基于门控机制的GCN模块

本小节提出了一种改进的图卷积模块门控图卷积（Gate GCN Network, GGCN），模块首先利用GCN进行卷积获取不同层节点表示，然后利用注意力门控机制融合不同卷积层的结果作为最终的节点表示，能够有效避免过度平滑的问题和保留不同抽象层次的特征，增加模型的鲁棒性，具体思路如下：图卷积每经过一次卷积之后，得到的特征越抽象，如果只取最后一次卷积的结果，这就会导致过度平滑，为了解决这个问题，我们使用门控机制融合不同抽象层次的特征，也就是每一次卷积之后的特征做融合。其思想来源于LSTM的门控机制，具体如图 20所示。其具体公式如（5-1），其中表示卷出层数，表示每一层的嵌入维度，表示最终的经过门控机制的节点特征。

 （5-1）

这种方法的优势在于，它不仅仅依赖于最后一层的卷积结果，而是将多个卷积层的信息有效结合起来。通过门控机制，GGCN可以动态地选择和调整不同卷积层的重要性，从而更好地保留不同抽象层次的特征。这种做法有效避免了可能出现的过度平滑问题，同时增强了模型的鲁棒性和表征能力。

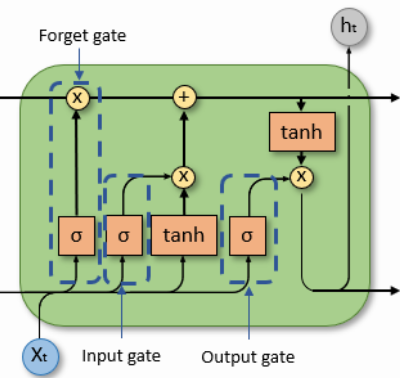


图 20 LSTM的门控机制

## 5.3 自适应图卷积

自适应图卷积（Adaptive Graph Convolution, AGC）是一种应用于图神经网络（GNN）的算法，它通过学习图结构中的边权重来增强模型处理图数据的能力。

在传统的图卷积网络（GCNs）中，节点的特征更新主要依赖于其邻居节点的特征聚合。然而，这种方法并没有考虑到不同边在信息传递中的重要性可能存在差异。例如，在社交网络中，某些联系可能比其它联系更为紧密或重要。自适应图卷积正是为了解决这一问题而提出的，它通过引入边权重来调整节点间的信息流动，使得模型能够更加灵活地捕捉图中的复杂结构和关系。

自适应图卷积的核心原理是为图中的每条边分配一个可学习的权重参数，该参数在训练过程中不断更新，以反映边在信息传输中的重要性。具体来说，AGC通过权重矩阵调整节点特征的聚合过程。在每次特征更新过程中，节点的新特征不仅包括其自身特征，还包括其相邻节点特征和相应边缘权重的加权和。这一过程可以表示为（5-2）：

 （5-2）

其中，是节点在第层的表示，是节点的全部相邻节点，是节点到的边权重，表示sigmoid激活函数。

自适应图卷积中的边权重更新是一个关键的环节。权重矩阵通常是通过一个可训练的神经网络来学习，该网络可以根据节点的特征和结构信息来预测每条边的权重。在训练过程中，边权重的更新依赖于损失函数的梯度下降，这使得模型能够自动调整权重以最小化预测误差。

## 5.4 基于混合视图与自适应门控图卷积的恶意URL检测模型

**5.4.1 模型结构**

适应门控图卷积的恶意URL检测模型如图 21所示，其由Bert预训练模型、自适应门控GCN模块以及最后的全连接层组成。

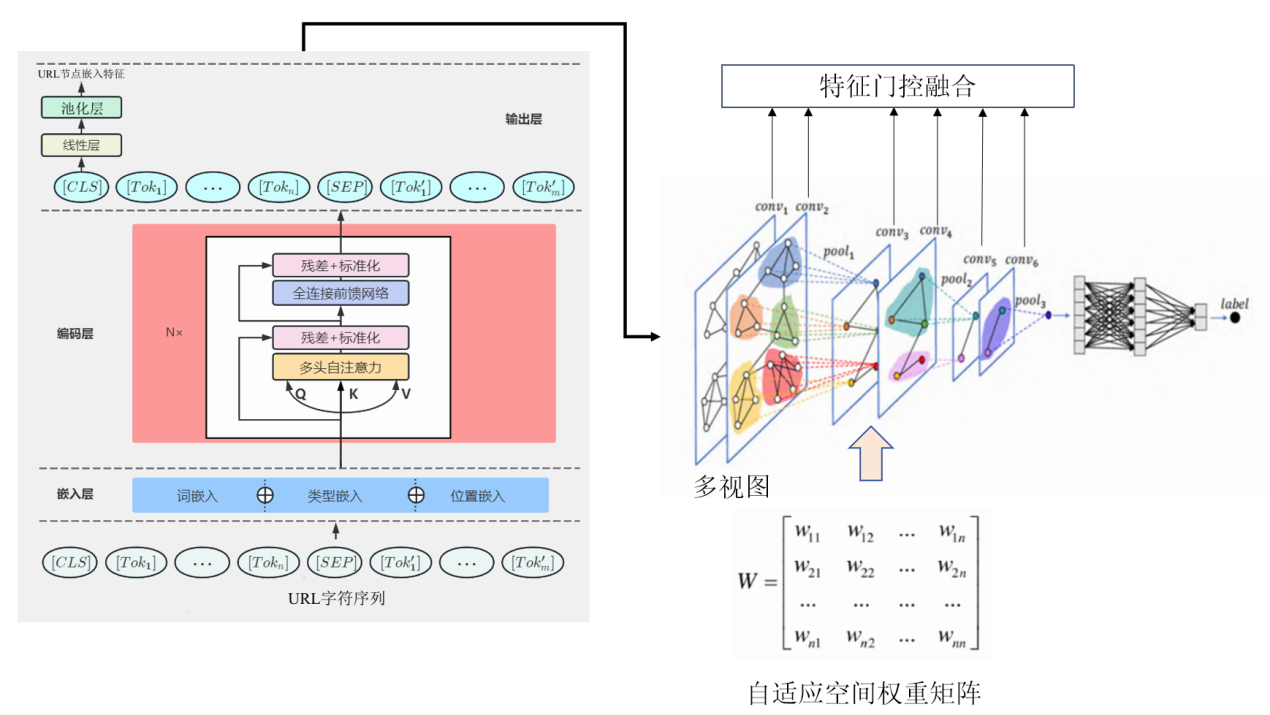


图 21自适应门控图卷积的恶意URL检测模型

自适应门控GCN模块在GCN的基础上引入了一个能够随梯度下降进行参数更新的空间权重矩阵，其代表着节点之间的相关性大小，在进行节点特征传递时用于限制信息的传递幅度。具体公式如（5-3）：

 （5-3）

此外，我们引入了特征门控融合机制，其主要思路为将每一层图卷积操作之后的节点向量进行特征融合，然后输出最后一层的节点表示，其能够保留不同层的特征，类似于残差连接，能够解决由于多次卷积造成的过度平滑问题。如下所示公式（5-4）。

 （5-4）

## 5.5 结果分析

### 5.5.1 数据集

同样采用ISCX-URL和Malicious\_phish URL两种数据集，从样本中选择80%的节点作为训练集集，剩下的20%做测试节点。训练时将测试节点进行MASK屏蔽，测试将训练集进行MASK屏蔽。

### 5.5.1 模型分类结果分析

图 22显示了模型训练过程中损失和Acc值的趋势，表明模型不断收敛，没有欠拟合或过拟合。

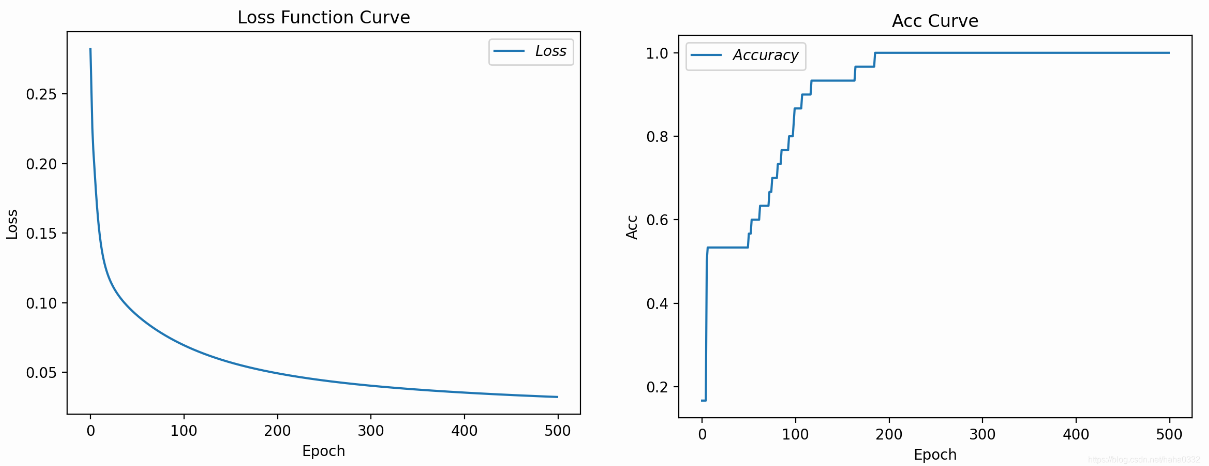


图 22 模型在ISCX-URL训练集训练过程Loss与Acc变化

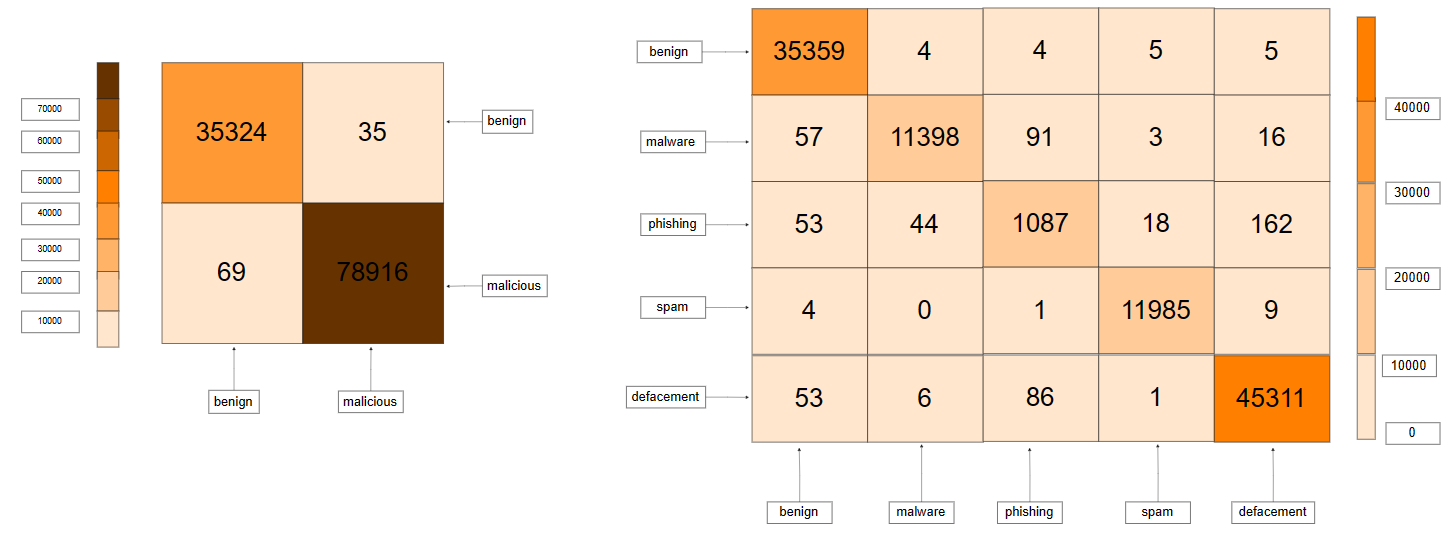


图 23 多视图自适应门控GCN模型在ISCX-URL数据集上二、多分类效果

图 23展示了多视图自适应门控GCN模型在ISCX-URL数据集上二、多分类效果。在二分类效果上，所提模型在ISCX-URL数据集上取得了99.97%得Precision，99.91%得Recall和99.94%的F1分数；在Malicious phish数据集上取得了99.42%得Precision，99.64%得Recall和99.53%的F1分数，证明模型取得了非常不错的预测精度，具体如表 16所示。

表 16 多视图自适应门控GCN模型在两种数据集上二分类的实验结果

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | Precision | Recall | F1-score |
| Malicious phish | 99.42% | 99.64% | 99.53% |
| ISCX-URL | 99.97% | 99.91% | 99.94% |

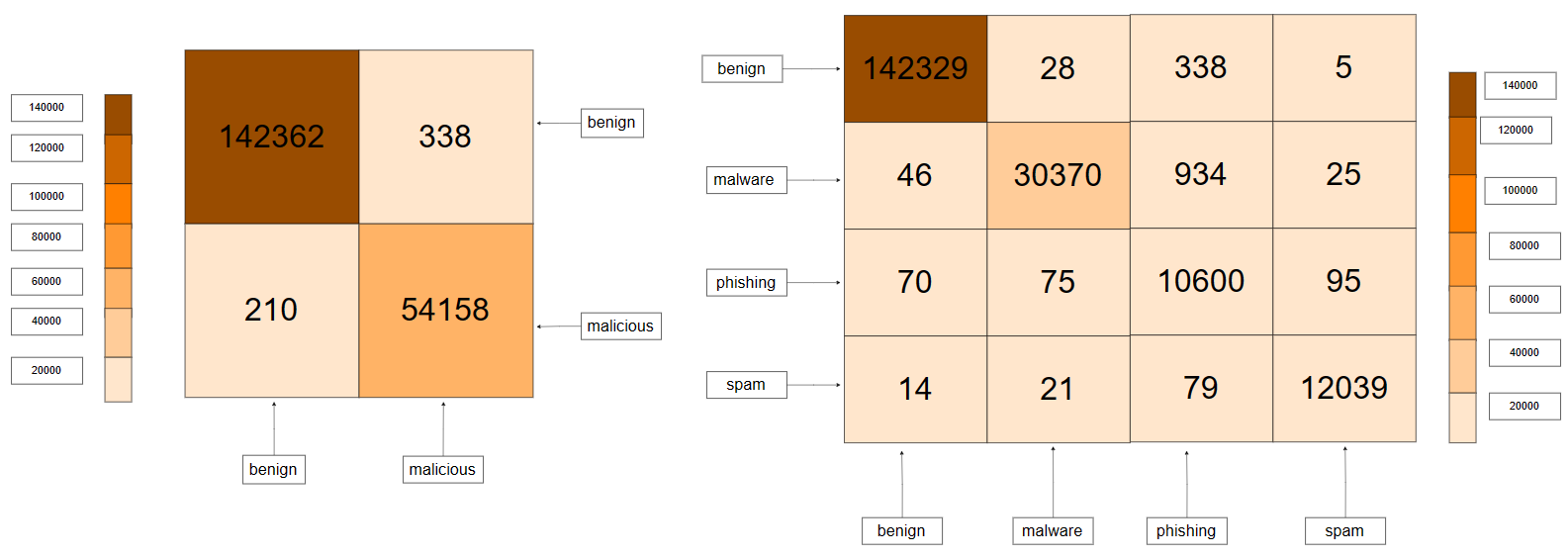


图 24 多视图门控GCN的恶意URL检测模型在Malicious phish数据上二、多分类效果

在恶意URL的细致分类的识别任务上，多视图门控GCN的恶意URL检测模型也取得了非常好的预测结果，各种指标都达到了99.50%以上，具体结果如表 17所示。

表 17 多视图自适应门控GCN模型在两种数据集上多分类的实验结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模型 | benign | phishing | malware | span | defacement |
| Malicious phish | 99.72% | 99.73% | 99.89% | 99.85% | 99.90% |
| ISCX-URL | 99.80% | 99.78% | 98.90% | 99.87% | 98.91% |

## 5.6 消融试验

本小节在“多视图门控GCN的恶意URL检测模型”的基础上进行消融实验，以探究所提出改进点的有效性；同时进行模型参数敏感性分析，以探究模型的鲁棒性。

### 5.6.1 参数敏感性分析

参数敏感性分析旨在设置不同的模型参数，例如学习率、批次大小、嵌入维度和训练轮次不同取值对模型预测精度的影响结果。在这里设置两种实验：1、不同学习率对模型的影响；2、不同训练轮次对模型预测精度的影响。

采用控制变量法每次只改变其中的某一个参数而控制其他参数大小不变，从而研究改变的参数对实验结果的影响。经实验验证，在ISCX-URL数据集上进行二分类，学习率大小以及Epochs大小对本文模型实验结果的影响如表 18和表 19所示，主要探究的指标为精确率、召回率、F1-score。

表 18 不同学习率对模型预测结果的影响

|  |  |  |  |
| --- | --- | --- | --- |
| 学习率 | 精确率 | 召回率 | F1-score |
| 0.1 | 99.54% | 99.65% | 99.59% |
| 0.01 | 99.84% | 99.85% | 99.84% |
| 0.001 | 99.97% | 99.91% | 99.94% |

设置Epoch为400，可以发现，当学习率较大时，模型预测精度有所减小，原因可能是模型在最优点来回震荡，却无法更近异步减小。当学习率变小，模型可以收敛到一个比较好的状态。

表 19不同Epoch模型预测结果的影响

|  |  |  |  |
| --- | --- | --- | --- |
| Epochs | 精确率 | 召回率 | F1-score |
| 200 | 98.87% | 98.91% | 98.89% |
| 300 | 99.96% | 99.91% | 99.93% |
| 400 | 99.97% | 99.91% | 99.94% |

从表 19中可以得出，在其他参数保持不变的情况下，且将学习率设置为 0.001时，Epochs值设置为400以上是最佳的。

### 5.6.1 去除自适应空间权重矩阵

此消融实验去掉了模型当中自适应空间权重矩阵模块，使模型无法学习URL节点之间边的权重关系。

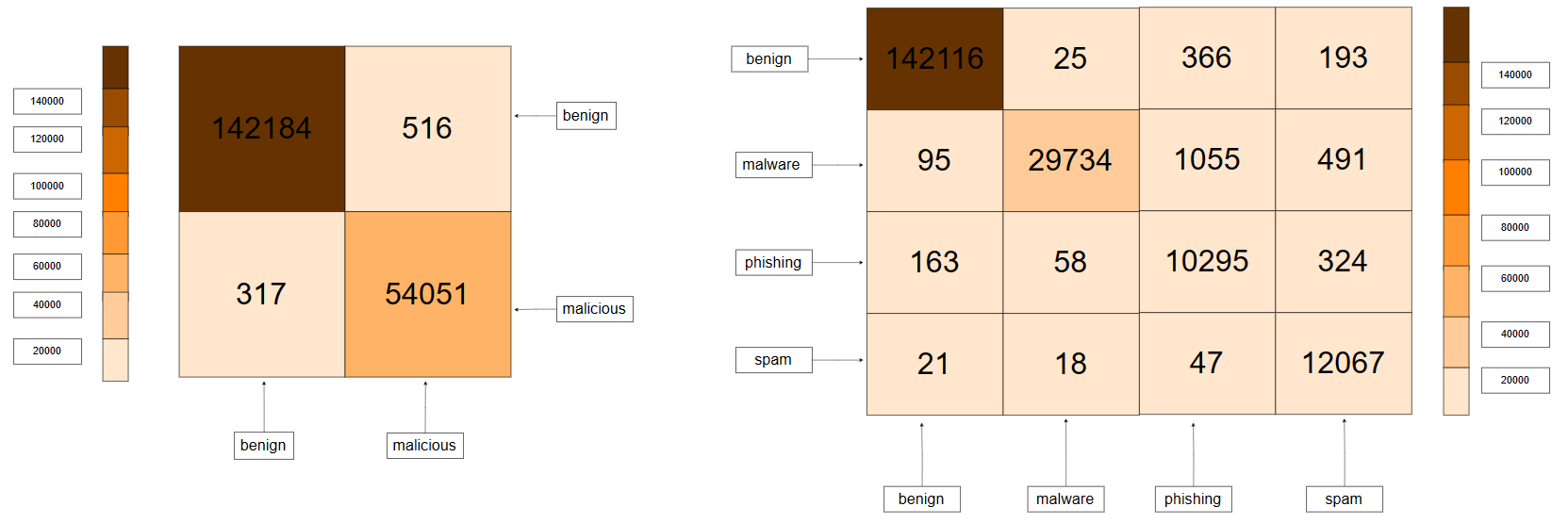


图 25多视图门控GCN在Malicious phish数据集上的分类效果

从图 25可以看出，在剔除空间权重矩阵之后，即“基于多视图和自适应门控GCN的URL检测模型”变成“基于多视图和门控GCN的URL检测模型”，从图中可以看出，模型的预测精度有所降低，对于一些较难分辨的节点无法进行有效区分。

### 5.6.2 去除特征门控融合机制

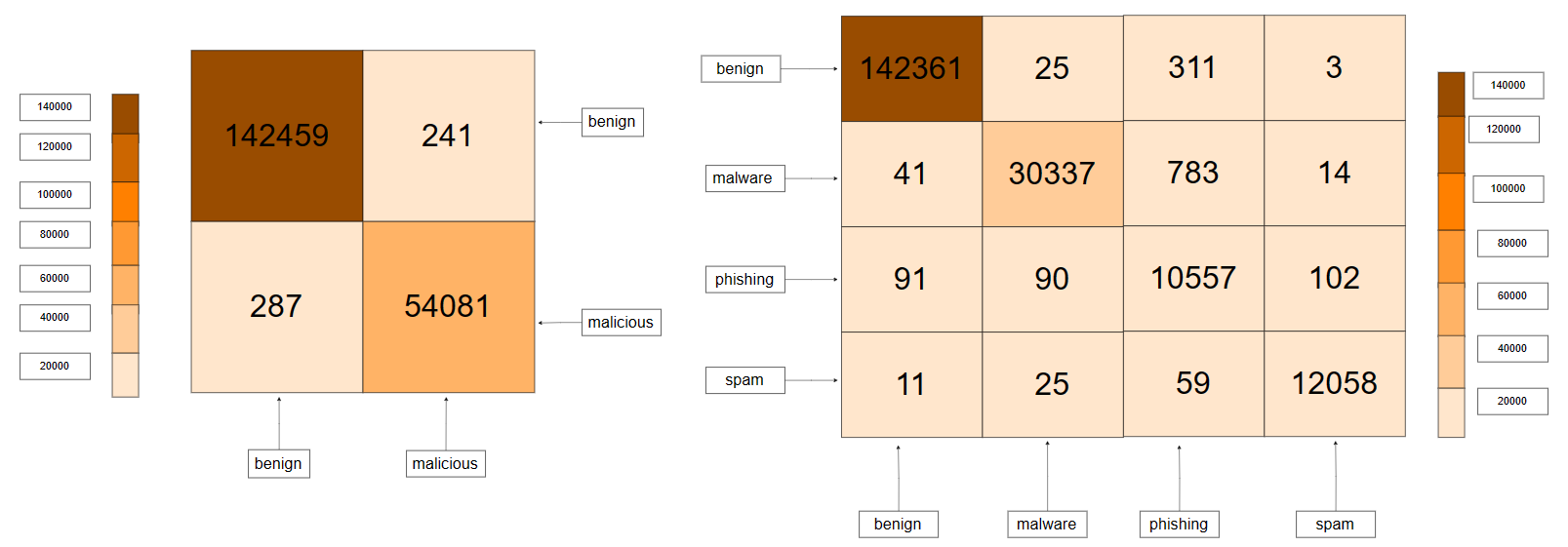


图 26多视图GCN在Malicious phish数据集上的分类效果

此消融实验去除了GCN网络层之间的门控融合部分，但是仍然保存自适应权重矩阵部分。从图 26可以看出，模型的效果也有一些下降，但是下降程度并没有去除自适应空间权重大。

综上所述，所提出的两个模块能够显著的提升模型的预测精度，充分证明了所提的考虑多视图、考虑自适应邻接矩阵和门控机制的有效性。

### 5.6.4 与基线模型进行对比

为了证明“基于多视图和自适应门控GCN的URL检测模型”模型拥有更佳的恶意URL预测性能，我们与一些传统的基线模型进行了对比，具体结果如表 20和表 21所示。

可以看到“基于多视图和自适应门控GCN的URL检测模型”相比于基线模型预测进度有显著的提升，即所提出的模块和考虑的因素能够为模型提供更多的信息，模型能够自动学习得到这些信息并用于恶意URL分类，证明了本文研究的正确性和合理性，在实际应用当中，本文可以提供一种较好的技术方法。

表 20四种模型对ISCX-URL检测具体类别精确率的结果对比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模型\类别 | benign | phishing | malware | spam | defacement |
| LSTM | 95.46% | 95.12% | 95.06% | 95.57% | 95.74% |
| Text-CNN | 96.19% | 95.72% | 96.38% | 96.67% | 96.40% |
| Bert-GCN-FT | 97.40% | 98.54% | 98.42% | 97.75% | 98.48% |
| 多视图自适应门控GCN | 99.80% | 99.78% | 98.90% | 99.87% | 98.91% |

表 21四种模型对Malicious Phish URL检测具体类别精确率的结果对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模型\类别 | benign | phishing | malware | defacement |
| LSTM | 94.63% | 94.66% | 93.89% | 94.29% |
| Text-CNN | 95.23% | 95.96% | 94.51% | 95.01% |
| Bert-GCN-FT | 97.39% | 97.59% | 97.70% | 98.01% |
| 多视图自适应门控GCN | 99.72% | 99.73% | 99.89% | 99.85% |

## 5.7 本章小结

在本章节，我们提出了基于多视图的门控自适应图卷积URL分类模型，具体的来说，我们依据URL间的引用关系、跳转关系、同源关系构建了3种网络视图。其次，提出了一种改进的GCN模块——门控GCN，模块首先利用GCN进行卷积获取不同层节点表示，然后利用注意力门控机制融合不同卷积层的结果作为最终的节点表示，能够有效避免过度平滑的问题和保留不同抽象层次的特征，增加模型的鲁棒。同时，添加了自适应空间权重矩阵，能够从数据当中自动学习得到不同URL节点之间的关联性。之后，我们对该模型的预测精度进行了评估，同时进行了消融试验，证明了所提模块和改进点具有有效性，能够显著提升模型的精度。

# 第六章 总结与展望

## 6.1 总结

伴随着网络的快速发展，许多不法分子将目光投向了网络，网络攻击的手段多种多样，然后多数人并没有网络安全意识，所以在用户使用计算机访问服务的过程中可能面对网络钓鱼或者恶意软件的下载等，影响用户的计算机的正常使用，后果严重的可能会造成资金损失或者隐私泄露。本文对恶意URL的检测技术进行改进探索研究，提出了基于Bert预训练模型和考虑URL多重关系视图网络恶意URL检测模型，以下是本文的工作内容总结

1．提出了基于Bert-FT（Bert·Based·Fine-Tuning Model）的恶意URL分类模型，其使用Bert作为URL字符串的特征提取骨干，并进行下游分类任务微调，相比于基线模型，实现了更优的恶意URL分类检测效果。

2．考虑了URL之间的引用关系，利用GCN实现不同URL节点之间的表示传递和聚合，最后利用节点的表征进行节点分类任务。试验结果表明，相比于单独的URL字符串表征，基于图卷积神经网络GCN的恶意URL分类模型能够取得更好的分类检测效果。

3．进一步考虑到 URL之间存在不同的视图，以及现有图卷积存在过度平滑的问题，提出了基于混合视图与自适应门控图卷积的恶意 URL 检测方法。首先提出了一种新的图卷积方式门控图卷积(Gate GCN)，Gate GCN首先利用图卷积获取不同卷积层的节点表示，然后利用门控机制融合不同卷积层的结果作为最终的节点表示，能够有效避免过度平滑的问题和保留不同抽象层次的特征，显著增加了模型的鲁棒性。同时消融试验结果表明，所提模型与所添加的模块能够显著提升模型在各种URL任务上的分类性能。

## 6.2 未来展望

对已做工作内容进行回顾，我们总结了目前研究工作中的不足，具体的不足与解决方案如下：

1. 对于一个URL节点，将更多的特征考虑到节点嵌入当中，例如JavaScript代码特征、HTML文档中超链接标签等特征。
2. 当URL视图网络过大，需要更多的储存资源和计算资源，之后的研究应该关注对计算资源的节省。
3. 当前任务是将恶意和非恶意二分类URL检测与恶意URL多分类任务进行分开，在之后的研究中可以进行多任务学习，考虑更多的Loss损失函数，加速模型的收敛速度。