doi :10.3969/j.issn.1671-1122.2023.09.001

基于双重图神经网络和自编码器的网络异常检测

秦中元，马楠，余亚聪，陈立全
（东南大学网络空间安全学院，南京 211189）

摘　要：图神经网络在网络异常检测领域中的应用大多集中于单点特征的提取，忽略了连续流量之间的关联性的特点，文章提出了一种基于双重图神经网络和自编码器的网络异常检测方法 DGCNAE。该方法首先对通信数据进行图构建和子图划分，然后将子图送入两层图卷积神经网络，分别对点和边进行特征提取，最后采用无监督学习方法对划分出的子图进行训练。通过对子图划分时间间隔和迭代次数进行迭代实验，得出效果最佳的子图划分时间间隔和迭代次数，并在 3 个典型数据集上与已有算法进行对比实验，实验结果表明，该方法具有更高的准确率和泛化能力。

关键词：异常检测；图神经网络；自编码器

中图分类号：TP309　**文献标志码：**A　**文章编号：**1671-1122（2023）09-0001-11

中文引用格式：秦中元，马楠，余亚聪，等 . 基于双重图神经网络和自编码器的网络异常检测 [J]. 信息网络安全，2023，23（9）：1-11.

英文引用格式：QIN Zhongyuan, MA Nan, YU Yacong, et al. Network Anomaly Detection Based on Dual Graph Convolutional Network and Autoencoders[J]. Netinfo Security, 2023, 23(9): 1-11.

Network Anomaly Detection Based on Dual Graph Convolutional Network and Autoencoders

QIN Zhongyuan, MA Nan, YU Yacong, CHEN Liquan
(*School of Cyber Science and Engineering, Southeast University, Nanjing* 211189, *China*)

Abstract: Considering the application of graph neural networks in the field of network anomaly detection mostly focused on the extraction of single point features, while ignoring the correlation features between continuous messages. This paper proposed a network anomaly detection method based on dual graph convolutional networks and autoencoders. This method first constructed the graph and divided the subgraph of the communication data,

收稿日期：2023-06-05
基金项目：国家重点研发计划 [2020YFE0200600]
作者简介：秦中元（1974—），男，河南，副教授，博士，CCF 会员，主要研究方向为智能终端安全、人工智能安全和无线网络安全；马楠（2000—），女，江苏，硕士研究生，主要研究方向为网络安全；余亚聪（1997—），男，浙江，硕士研究生，主要研究方向为可信计算、密码学；陈立全（1976—），男，广西，教授，博士，CCF 会员，主要研究方向为移动信息安全、物联网系统与安全、云计算和大数据安全。
通信作者：秦中元 zyqin@seu.edu.cn

then sent the subgraph into the two-layer graph convolution neural network to extract the features of points and edges respectively, and finally used the unsupervised learning method to train the divided subgraph. In the experimental part, through the iterative experiment on the subgraph division time interval and iteration times, the subgraph division time interval and iteration times with the best effect were obtained. Comparative experiments with traditional algorithms on three data sets showed that our scheme is more accurate and has stronger generalization.

Key words: anomaly detection; graph neural network; autoencoder

0 引言

为了得到更高的传输速率，智能设备一般采用TCP/IP协议和基于以太网的技术进行数据传输，这在降低成本、提高速度的同时，使得数据暴露在互联网中，容易受到网络攻击。入侵检测系统（Intrusion Detection System，IDS）作为对防火墙的补充，在网络层的防护中起着重要的作用。入侵检测系统根据预先设计好的规则，对途经的网络报文进行数据分析，发现潜在的攻击意图，并及时响应，尽最大可能保证系统信息的完整性、安全性和机密性。目前，随着入侵检测系统和其他领域学科的发展，入侵检测系统不再是独立的一个分支，而是与其他学科相互借鉴、不断完善，如专家学习[1]、统计学习[2]、深度学习[3]等。

现代社会，网络攻击频率越来越高，攻击类型越来越丰富，这些攻击往往具有更高的隐蔽性和危害性。随着深度学习的高速发展，它已经深入各个领域，其中也包括网络安全领域。深度学习具有强大的学习能力，可以学习到数据中隐藏的高级特征，从而能够准确识别那些隐蔽的攻击并给出高置信度的判断。深度学习目前已经被广泛运用在入侵检测领域[4]，并且取得了非常不错的效果。但是目前存在入侵检测算法大多关注于流量的内容特征，导致不同网络流量之间的关联容易被忽略，无法抓住流量结构上的特征。复杂网络属于拓扑结构，对于一个网络节点而言，每时每刻都有流量数据的到来，而流量数据之间并非是独立的，往往具有一定的关联，流量数据之间的关联性往往也包含了入侵在结构上的特征。同时随着安全技术的发展，网络攻击等异常事件往往发生在多个设备间，网络流的单点分析已不足以及时发现异常，异常检测的精确度正逐渐下降。

图结构具有极强的表达能力，吸引了很多学者对如何将图结构应用于异常检测进行深入研究，并且取得了一定成果。图结构由节点和边组成，节点之间的联系通过边来表示。图结构可以将许多其他数据结构无法表示的情况进行抽象，使得计算机可以对这些复杂的关系进行分析。在通信网络中，两个节点之间的数据通信可以抽象成边，正常的通信可以抽象成正常边，而异常的通信则被判定为异常边。通过对图结构进行异常挖掘，可以及时阻止计算机网络中具有攻击行为的通信[5]，对防御网络攻击具有重要作用。图神经网络可以处理网络异常检测复杂庞大拓扑结构的图形数据，其中包括网络拓扑结构和节点行为分析，同时还可以捕捉节点的相似性与差异性，在保留重要信息的同时减少数据的维度。因此，本文提出了一种基于双重图卷积神经网络和自编码器的异常检测算法（Dual Graph Convolutional Network AutoEncoders，DGCNAE），首先通过对不同数据集依据不同规则进行子图划分，然后通过双重图神经网络分别对节点信息和边信息进行特征提取，最后通过自编码器的重建误差对流量进行异常划分。

本文具体贡献有以下3点：

1）DGCNAE算法对不同数据集按照不同的规则进行子图划分，将规则内的一段网络流量整合成包，并进行子图的构建，使得子图不仅具有数据包的内容特征，还具有结构特征。

2）传统图神经网络多用于对节点特征进行提取，却忽略了边特征。在通信网络中，两个节点之间的连接特征表现在边特征中，DGCNAE算法中，通过双重

图卷积算法分别对节点和边进行特征提取，并进行拼接，使得最终得到的特征包含更完整的信息。

3）自编码器可以通过对正常样本进行训练，做到无监督检测，大大减少了实验中对样本进行标注的时间成本，并且双重图卷积和自编码器结合的DGCNAE算法在准确率方面优于传统算法。

1 相关工作

国内外学者针对网络异常检测展开了大量的研究，网络异常检测一般可以分为基于特征的入侵检测和基于异常的入侵检测。基于特征的入侵检测通常利用已知的规则和标准来分类和处理网络流量；基于异常的入侵检测通过机器学习等技术对正常流量模型进行建模，当系统检测到与模型不匹配的行为时采取相应的行动进行阻止。但是无论是基于特征的入侵检测，还是基于异常的入侵检测，大多都使用文本或特征序列，它们的关注点都在流量本身的特征上，却忽略了多条流量之间的关联性和结构特征，同时也缺乏对多条流量关联性进行有效建模的方法[6]。图结构可以将网络中的拓扑结构抽象为节点和边的拓扑结构，将网络流量等信息表示为节点和边的属性，然后对网络节点之间的依赖关系进行建模分析，同时可以动态地更新和适应网络中的变化。图结构的引入为网络流量之间的建模提供了新的方向。

基于图的异常检测是指在大量图表示的数据中发掘异常的模式，目前已被广泛应用于网络流量检测，如物联网大数据异常检测、日志异常检测等。YAO[7]等人提出了DeepGFL框架，该框架通过网络流量构建具有多属性的属性网络流图结构，并利用深度图表征算法从网络流的原始特征开始，通过增加复杂性，学习更深层更有判别性的特征。最后训练随机森林分类器进行二分类，在真实数据集IDS2017上取得了准确有效的成果。针对网络中的高级持续性威胁（Advanced Persistent Threats，APT），FANG[8]等人利用图嵌入技术生成内部网络流量和日志的异构图，提出了使用异构图来构建横向移动路径的方法，该方法在直接追踪攻击者活动、解决网络中问题的同时保留路径关系，供后续安全分析所用。上述文献所采用的图表征学习技术存在对大规模和复杂图结构难以处理、数据丢失、处理能力有限等问题，随着神经网络的高速发展，能够更好处理图结构数据和自学习能力强的图神经网络技术已被广泛运用到异常检测中。

图神经网络主要分为4类[9]：图递归神经网络、图卷积神经网络、图自动编码器以及时空图神经网络。这些神经网络能够利用拓扑模式进行训练和测试，通过节点、边之间的消息传递机制来充分利用图结构，从而能够有效地学习和泛化基于图的数据，并以低维向量输出[10]。2016年，KIPF[11]等人首次提出图卷积神经网络（Graph Convolutional Network，GCN），他们在深层图神经网络的架构上加入了多层卷积层。图卷积就是指图中的每个节点都受到其他节点的影响而改变自己的状态直到最终的平衡。GCN相较于传统技术可以综合利用节点属性和图结构的多重信息，更适合处理具有局部领域结构的图形数据。自此，图卷积神经网络开始了高速发展。2020年，刘杰[12]等人设计了一种基于图卷积神经网络的工控网络异常检测算法，通过图卷积神经网络对数据进行图节点状态向量获取，并通过*k*-means算法对数据集进行聚类，实验证明该算法具有优秀的聚类效果和鲁棒性。TANG[13]等人将电力系统中异常流量检测转为半监督节点分类任务，将异常IP之间的交互关系进行图建模，引入经典图卷积网络，综合挖掘属性和结构信息，用于检测不同类型的攻击，他们所提出的方法与传统图聚类方法相比更加充分地利用了异常IP的属性与结构信息，取得了更优异的检测结果。图卷积神经网络在网络异常检测领域的应用证明了其在处理图模型具有独特性和优越性，更容易把握不同流量之间的关联性。

分析图的属性和结构可以将图的异常类型[14]分为异常节点、异常边、异常子图和异常事件。异常节点与其他节点相比通常含有不规则的特征表示。异常节点可依据结构信息进一步划分为全局异常、结构异常

和社区异常。全局异常指异常节点的属性与其他节点在整个图中的属性存在显著差异；结构异常是指异常节点的拓扑结构与其他节点的差异较大；社区异常是指在一个图中，存在一些社区（子图）内部的节点和边的连接模式与整个图其他部分不同，表明社区中存在异常[15]。异常边与异常节点类似，异常表示为与其他边相比具有不规则的特征表示，通常异常节点和异常边会使用异常分数进行衡量。异常子图是以图为单位进行衡量，更关注整个图的异常表示，其中包括节点异常表示和边异常表示。异常事件则是通过对比异常事件与其他事件在时间序列上的不同，常用于动态异常检测。

目前大部分网络异常检测都是基于节点的，单独的基于节点的检测往往只考虑节点本身属性而忽视节点间的关系，为了提高网络异常检测的准确率，已有学者利用边辅助节点或者将两者相结合的方法进行检测。DING[16]等人在物联网的活动节点上进行分布式流量异常检测，融合了图神经网络与两个多层感知机。其中，边的图多层感知机用于分类和预测邻接节点的异常概率，节点图多层感知机用于产生并更新自己异常状态的概率。实验证明该方案的训练和预测时间得到提升，引入的注意力机制使准确率和精度得到提高，为物联网活动节点的安全保障提供了有效解决方案。ZOLA[17]等人通过合理定义节点和边概念，将原始数据集分割为固定的时间间隔，从时间序列数据中提取网络流量图特征，结合边信息表征每个节点，突出网络的微动态。考虑到正常通信流量占大多数，即使是时序数据流量图也会存在类图不平衡的情况，ZOLA[17]等人分别用无监督模型直接评估不平衡数据集和在改进的有监督机器学习中引入了两种有关图数据的预处理技术，利用碎片化的时序数据减少类图不平衡性。LO[18]等人改进了GraphSAGE算法的输入、聚合器函数以及输出部分，在考虑节点特征的同时进行边缘特征嵌入。其主要思路是：设置当前节点的信息传递是所有采样邻域中的边特征的均值；节点的嵌入特征是当前节点与其上一层的信息经过聚合、计算之后的结果；边嵌入特征即两个节点嵌入特征的聚合。经过多次迭代，不断聚合邻域，直到每个节点和边几乎都包含全局信息的聚合。文献[18]中对所提出的方案进行了多项实验评估，不管是在二分类还是多分类中都取得了较好的效果。

在本文中，主要针对不同数据集在子图层面的异常进行研究，考虑通信网络构成的图中，每条边代表两个节点之间的信息流交互，出现的异常往往体现在边上，而不是节点上，因此边特征比节点特征更加丰富，研究的关键在于如何对节点异常和边异常进行量化和融合。

按照数据集是否有标签可以将训练分为有监督学习和无监督学习，有监督学习中的训练样本由输入向量和期望输出的类别即标签构成。有监督学习训练的网络准确率较高但需要花费大量时间对数据进行额外标注，并且难以检测到新型攻击[19]。而无监督学习无需对数据集进行标注，不过效果很难评估。自编码器（Auto-Encoder，AE）是一种无监督的学习模式，它利用了反向传播算法作为前馈训练，目的是用最小的畸变重构输入[20]，训练的过程中需要设计一个损失函数，让编码器的输入和输出尽可能相似。为了减少子图标记的时间消耗，本文在异常识别模块采用自编码器对划分后的子图进行训练。

2 方法描述

在图模型的异常检测中，仅仅使用传统的分类和聚类算法无法做到异常的准确识别，而使用普通的神经网络则无法对图模型进行处理。本文提出了基于双重图卷积神经网络和自编码器的DGCNAE算法，由于异常出现的位置可能在点上，也可能在边上，本算法通过图神经网络对通信数据包划分出的子图进行点属性和边属性的双重特征提取，进行融合后得到子图的属性和结构特征，并送入自编码器进行异常分类训练，具体架构如图1所示。该模型共分为3个部分，分别是预处理模块、双重卷积神经网络、异常检测模块。下

面分别对3个部分进行介绍。

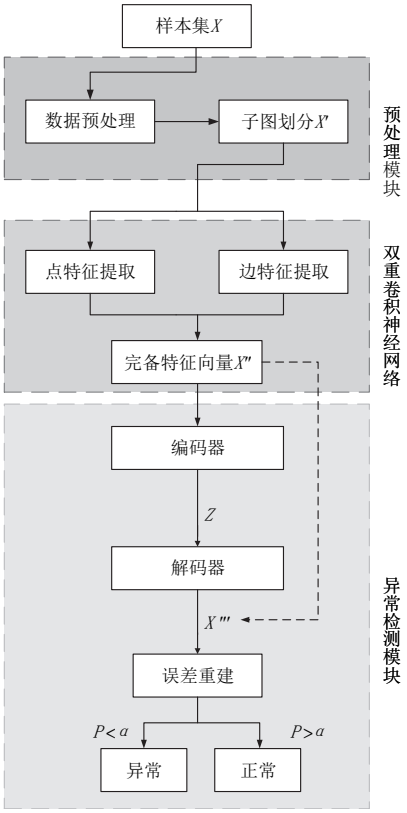

图 1 DGCNAE 模型

2.1 预处理模块

预处理模块负责对原始样本集进行预处理工作，包括样本处理和子图划分。先利用已开源的CICFlowmeter工具对流量包进行特征提取生成所需样本。在样本处理过程中，需要对样本进行数据筛选、数值转化、归一化等工作。在子图划分过程中，需要对样本集图模型建模，对节点特征和属性特征进行划分，得到全图的邻接矩阵，并根据子图划分准则进行子图划分，得到子图数据集。预处理流程如图2所示，具体流程如下。

1）脏数据剔除：该步骤对数据集中的脏数据进行处理，脏数据包括Nan和Infinity数据，需要对含有这两种数据的行进行剔除。

2）数据处理：该步骤对每条数据进行数据处理，包括字符型特征的数值化和数值特征的归一化。由于模型无法对字符型特征进行学习，需要使用One-hot编码，将字符型特征转换为数值特征。由于数据中某些

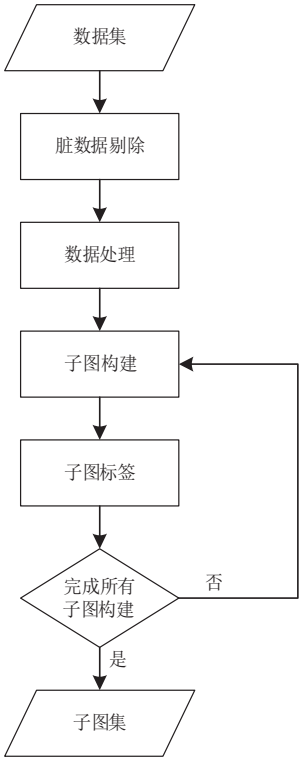

图 2 子图预处理流程

数值型特征跨度大，需要对数值特征进行统一的归一化操作，以避免个别极端数值对模型训练的影响。

3）子图构建：该步骤将每条数据按照时间戳进行子图的构建。以IDS2017为例，该数据集的跨度为一天中1：00至12：59共计12 h，取1 min内所有的数据报文作为子图划分的最小单位。每条数据报文包含了源IP、目的IP、报文异常类型以及其他数据特征，在子图构建的过程中，每个IP节点作为一个点，记录与其相连的其他IP节点通信情况，如存在通信行为，则两个节点之间用边进行连接。鉴于不同数据集单位时间内到达的信息量不同，需要进行实验以确定最合适每个数据集的子图划分时间。

4）子图标签：该步骤对构建完的子图进行标签的标注。在IDS2017数据集中，规定当子图中的异常边数超过300时，标注为异常图；否则，标注为正常图。

2.2 图卷积神经网络

图是由一系列顶点和一系列边构成的一种特殊的数据结构，可以用 $G=(V,E,A)$ 表示，其中 V 表示图

中点的集合，$V=(v_1,v_2,\cdots,v_n)$。E表示边的集合，每条边可以表示为 $e_{i,j}=(v_i,v_j)$。$\boldsymbol{A}$表示该图的邻接矩阵，大小为$N\times N$，代表了图中节点之间的连通情况，其中 $A_{i,j}=w_{i,j}$，当 $w_{i,j}=0$ 时，表示点i和点j之间没有边连接；当 $w_{i,j}\neq 0$ 时，表示点i和点j之间的连接权重。在本文的研究目标下，下面列出较为常见的5种图结构。

1）有向图：节点之间的边具有方向，箭尾表示信息的发送者，箭头指向信息的接收者。在实际网络流量数据中，数据包存在源IP和目的IP，体现在图中，用有向图代表信息流的传递过程更加合理。

2）权重图：节点之间的边具有权重信息，用于定量地表示关系之间的重要程度。在本文背景下，节点之间的通信交流重要程度无法使用合适的标准进行衡量，因此当两个IP之间存在交流时，权重统一定为1，这样可以简化训练过程的复杂性。

3）边异构图：节点之间的边可以使用不同的结构，如边的方向、权重、类型等。在复杂的网络通信图中，不同IP之间既可以单向交流，也可以双向交流。

4）点异构图：在点异构图中，节点可以分属于不同类型的图结构。在复杂的网络通信图中，不同的节点所处的网段不同，各个网段之间的图结构各异。

5）时空图：随着时间序列的推进，同一个图中的点、边可能都会发生变化。在实际情况中，同一对IP之间，可能时而通信，时而隔绝，节点的属性状态也各不相同。为了处理时间对图的影响，引入时间序列，可以更好地动态处理时序类型任务。

在图卷积网络提出之前，卷积神经网络就常用于提取图像的特征，它使用一个卷积核在图像上移动，图像在结构上通常具有平移不变性，即图像内部结构不随卷积核位置的变化而变化。图卷积神经网络是在前者基础上的延伸，在处理图信息时也采用权重共享机制，但不使用固定大小的卷积核而是同一子集内的节点使用同一卷积核。

本文采用图卷积神经网络负责对子图的特征提取，包括节点特征提取和边特征提取。模型中采用双重图卷积神经网络，处理对象分别为子图的点集和边集，最后对提取出的节点特征和边特征进行线性变换，拼接出该图属性和结构的完备特征。

在点特征提取阶段，使用经典GCN[11]中基于近似图谱核的逐层传播规则对图中的点进行特征提取，如公式（1）所示。

$$\boldsymbol{H}^{(l+1)}=f(\boldsymbol{H}^{(l)},\boldsymbol{A})=\delta(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{\frac{1}{2}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}) \quad (1)$$

其中，$\boldsymbol{H}^{(l)}$ 表示第 i 层的特征表达，对于输入层而言，原始特征向量 $\boldsymbol{X}$ 为初始特征表达，即 $\boldsymbol{H}^{(0)}=\boldsymbol{X}$。$\boldsymbol{A}$ 表示该图的邻接矩阵，包含该图节点之间的连接信息。$f(\cdot)$ 表示传播函数，输入为上一层特征表示 $\boldsymbol{H}^{(l)}$ 和邻接矩阵 $\boldsymbol{A}$。由于邻接矩阵 $\boldsymbol{A}$ 对角线的值均为 0，即节点和自身之间没有连接，这会导致在传播的过程中，忽略节点本身的特征，故需要对 $\boldsymbol{A}$ 进行处理，使其不仅具备邻接节点的信息，也具备自身的信息。

首先在邻接矩阵$\boldsymbol{A}$的基础上加上单位矩阵$\boldsymbol{I}$，这样就可以使得对角线元素全为1，如公式（2）所示。然后使用公式（1）中的 $\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{\frac{1}{2}}$ 对处理后的邻接矩阵 $\tilde{\boldsymbol{A}}$ 进行归一化处理，避免改变传播过程中特征原本的分布。其中，$\tilde{\boldsymbol{D}}$ 为 $\tilde{\boldsymbol{A}}$ 的矩阵，$\boldsymbol{W}^{(l)}$为第$\boldsymbol{l}$层选取的随机参数，$\delta(\cdot)$为激活函数。

$$\tilde{\boldsymbol{A}}=\boldsymbol{A}+\boldsymbol{I} \quad (2)$$

考虑网络通信中边代表两节点之间的信息交互，更容易存在异常，边特征比节点特征更加丰富，需要在常规的节点特征提取前提下，将图进行变换，增加对边特征的提取过程。由于每条边存在源节点和目的节点，当两条边存在相同的源节点或目的节点时，判定这两条边为相邻边，即可构建出边的邻接矩阵，具体判定准则如公式（3）所示。

$$\boldsymbol{A}_E^{i,j}=\begin{cases}1, \boldsymbol{e}^i_{source}=\boldsymbol{e}^j_{source} \text{ 或者 } \boldsymbol{e}^i_{des}=\boldsymbol{e}^j_{des}\\ 0, \text{其他}\end{cases} \quad (3)$$

其中，$\boldsymbol{e}^i_{source}$ 表示第 i 条边的源节点，$\boldsymbol{e}^i_{des}$ 表示第 i 条边的目的节点，$\boldsymbol{A}_E^{i,j}$ 为边的邻接矩阵。相比于点的邻接矩阵以 IP 节点为单位，记录 IP 节点之间的连接情况，

边的邻接矩阵以边为单位，记录两条边之间的连通情况。通过对公式（1）进行改进，得到关于边特征的传播方式如公式（4）和公式（5）所示。

$$\boldsymbol{H}_E^{(l+1)} = f(\boldsymbol{H}_E^{(l)}, \boldsymbol{A}) = \delta(\tilde{\boldsymbol{D}}_E^{-\frac{1}{2}} \tilde{\boldsymbol{A}}_E \tilde{\boldsymbol{D}}_E^{\frac{1}{2}} \boldsymbol{H}_E^{(l)} \boldsymbol{W}_E^{(l)}) \tag{4}$$

$$\tilde{\boldsymbol{A}}_E = \boldsymbol{A}_E + \boldsymbol{I} \tag{5}$$

为进一步说明将子图进行线图转换的流程，图3表示了一个子图如何将边以节点的形式构成一个新图，在该例子中，原始图的点邻接矩阵 $\boldsymbol{A}$ 由公式（6）所示，经过转换后线图的边邻接矩阵 $\boldsymbol{A}'$ 由公式（7）所示。图3中 a、b 两条边具有相同的目的节点，两条边依据公式（3）被判定为相邻边，同理 c、d 边有着相同的源节点，c、e 边有着相同目的节点，cd 和 ce 被判定为相邻边。

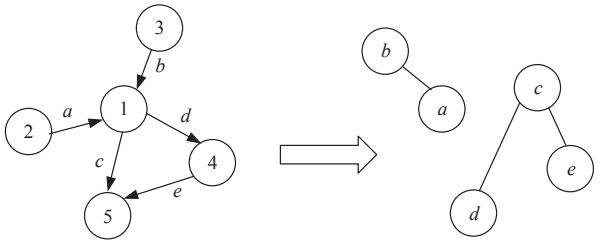

图 3 线图转换流程

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{6}$$

$$\boldsymbol{A}' = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{7}$$

2.3 异常检测模块

异常检测模块负责对压缩后的子图特征进行异常识别。根据子图的异常边数量，可以得到人为划分的子图标签，送入异常检测模型进行训练。测试阶段根据异常检测模型输出的异常分数进行异常划分，异常分数大于阈值的样本被判定为异常数据。在对数据集进行子图划分和双重GCN后，得到了完备的图表示向量和标签，即可以使用基于数据流的异常检测算法对样本集进行异常识别。本文选择自编码器AE[21]作为异常识别的模型，只需要使用正常的子图作为训练样本。使用AE进行异常检测的模型如图4所示。

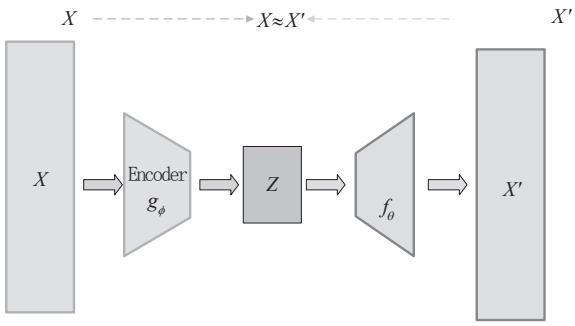

图 4 自编码器模型

由图4可以看出，AE对输入的图向量进行编码后，希望经过编码解码后的向量接近输入向量，即在多维空间中的距离尽可能地接近。因此，本文使用重建误差对样本进行异常判别，重建误差的表示如公式（8）所示。在训练阶段使用正常数据对自编码器进行训练，并在训练过程中计算正常数据子图经过AE编码解码之后的误差范围，取其最大值作为异常分数阈值。在测试阶段，当输入样本的重建误差大于阈值时，则判定为异常样本，否则为正常样本。

$$Reconstruct\_Score = ||Decode(Encode(X)) - X||_2 \tag{8}$$

2.4 整体算法

本文提出的异常检测模型可以将通信网络数据进行子图划分，首先对每个子图进行图建模，然后通过图卷积神经网络进行属性和结构的特征提取，最后通过异常检测算法进行异常判别。与传统的异常检测算法相比，该模型不仅对节点的属性结构进行了提取，还加入了对节点间结构的特征提取，最后通过自编码器模型的使用，引入无监督学习对子图进行学习，减少了对子图标记的时间消耗。本节中对该算法各个模块进行描述，以下是DGCNAE模型的伪代码。

算法1 DGCNAE算法

Input: Communication data set X, subgraph time interval T
Parameters: Reconstruction error α
Output: the labels of test dataset Y_{test}
1 SubgraphNum ←(the time interval of X)/T
2 for i in SubgraphNum:
3 Construct Subgraph $\boldsymbol{X}^i$
4 if the number of abnormal edge of $\boldsymbol{X}^i$>300
5 $Y^i \leftarrow 0$
6 else $Y^i \leftarrow 1$
7 end if
8 $\boldsymbol{X}^i_{node}$, $\boldsymbol{X}^i_{edge} \leftarrow$ Node and edge feature of $\boldsymbol{X}^i$
9 $\boldsymbol{A}^i$, $\boldsymbol{A}^i_{edge} \leftarrow$ Node and edge adjacency matrix of $\boldsymbol{X}^i$
10 $\widetilde{\boldsymbol{A}^i} \leftarrow \boldsymbol{A}^i + \boldsymbol{I}$
11 $\widetilde{\boldsymbol{A}^i_E} \leftarrow \boldsymbol{A}^i_E + \boldsymbol{I}$
12 Calculate node features $\boldsymbol{H}^i_{node} = ReLU\left(\widetilde{\boldsymbol{D}^i}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}^i}\widetilde{\boldsymbol{D}^i}^{\frac{1}{2}}\boldsymbol{X}^i_{node}\boldsymbol{W}\right)$
13 Calculate edge features $\boldsymbol{H}^i_{edge} = ReLU\left(\widetilde{\boldsymbol{D}^i_E}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}^i_E}\widetilde{\boldsymbol{D}^i_E}^{\frac{1}{2}}\boldsymbol{X}^i_{edge}\boldsymbol{W}\right)$
14 $\boldsymbol{H}^i \leftarrow concat\left(\boldsymbol{H}^i_{node}, \boldsymbol{H}^i_{edge}\right)$
15 end for
16 for j in the length of train dataset X_{train}:
17 Train Auto-encoder
18 end for
19 for k in the length of test dataset X_{test}:
20 $error^k \leftarrow \left\| Decoder\left(Encoder\left(X^k\right)\right) - X^k \right\|_2$
21 if $error^k$>α $Y_{test} \leftarrow 0$
22 else $Y_{test} \leftarrow 1$
23 end if
24 end for
25 return Y_{test}

在上述伪代码中，第1~7行表示子图构建的过程，其中SubgraphNum 表示子图的预设数量，由数据集时间跨度和子图划分时间决定。对于每个划分出的子图 $\boldsymbol{X}^i$ 将得到对应的标签 $\boldsymbol{Y}^i$ 。第8~15行表示子图特征提取过程，$\boldsymbol{X}^i_{node}$ 和 $\boldsymbol{X}^i_{edge}$ 分别表示节点和边的属性特征，$\boldsymbol{A}^i$ 和 $\boldsymbol{A}^i_{edge}$ 分别表示节点和边的邻接矩阵，第12~13行表示节点和边的特征向量获取的过程，对分别得到的特征进行拼接，即可得到完备特征表示$\boldsymbol{H}_i$。第16~17行表示模型的训练过程，将使用自编码器进行训练。第19~25行表示模型的测试过程，如果重建误差 $error^k$ 大于阈值α，则将该样本标记为异常样本，反之则为正常样本。

3 实验分析

3.1 相关数据集

本文实验的目的是验证提出的DGCNAE模型在处理拥有结构特征的数据集上的优越性，因此本实验采用的数据集需要包含除了常规的数据特征外的独特结构性，分别采用了IDS2017、Reddit数据集和Digg数据集，以下分别对3种数据集进行介绍。

1）IDS2017数据集

IDS2017数据集是由加拿大安全研究所通过网络攻击模拟收集到的数据，该数据集中每条数据包括源节点IP、目的节点IP、源端口号、目的端口号、时间戳等75个特征，每条数据都存在标注好的标签。周三的网络流量数据最多，因此选取周三的数据作为本实验的数据集，该数据集中包括正常数据标签“BENIGN”和“DoS GoldenEye”、“DoS Hulk”、“DoS Slowhttptest”、“DoS Slowlories”、“Heartbleed”5种异常标签。共计692703条数据，包含的IP地址数量为9015个，正常数据和异常数据的比例为2∶1，将该数据集构建成图模型，则该图的节点数为9015个，边数为692703条。

2）Reddit数据集

Reddit是Condé Nast Digital公司的一个社交新闻站点，用户可以在该网站上发布自己的原创帖子或者链接到别的帖子，对帖子发表看法。Reddit数据集采集了2014年1月到2018年8月期间不同话题之间的链接，每条链接都可以抽象成一条边，链接的起点和终点都是帖子，每条边都具有积极和消极两种情感倾向作为标签，实验中采用一天的超链接构建子图，则该图总节点数为55863个，边数为858490条。

3）Digg数据集

Digg数据集的数据来源于美国在线新闻分享网站Digg，用户可以在线对新闻进行阅读和评论，Digg数据集由用户的发帖和回帖数据构成，构成一个巨大的社交网络，抽象成图模型后，共包含30360个节点和

85155 条边。由于该数据集中不包含异常数据，故采用异常注入[22]的方式，随机选择20%的子图进行异常注入。3 个数据集的节点数量、边数量和标签情况如表 1 所示。

表 1 相关数据集基本情况

数据集	节点数量 / 个	边数量 / 条	是否含有标签
IDS2017	9015	692703	是
Reddit	55863	858490	是
Digg	30360	85155	否

3.2 实验参数设置

实验采用的硬件设备为拥有 Intel Core i7 处理器和 GTX3060 显卡的计算机。软件环境方面，本文使用 Python3 环境和 PyTorch 库进行图神经网络和自编码器的仿真实验。仿真实验的初始化参数如表 2 和表 3 所示。

表 2 图神经网络参数

参数	取值	参数含义
Epochs	200	训练轮次
Seed	40	种子数
Hidden_units	16	隐含层单元数量
Dropout	0.5	随机失活率
Learning rate	0.001	模型的学习率
Weight_decay	5e−4	权重衰减系数

表 3 自编码器模型参数

参数	取值	参数含义
Input_dim	32	输入 X 的维度
Latent_dim	10-2-10	隐含层的维度
Train_iter	500	迭代次数
Learning rate	0.001	模型的学习率
Batch-Size	32	每次训练的数据量
Sigmoid	ReLU	激活函数
Optimizer	Adam	优化器

3.3 实验结果分析

实验结果从实验参数设置对实验结果的影响和不同算法之间的对比实验两个角度进行分析。

为了探究实验参数对实验结果的影响，本节使用子图划分时间和迭代次数作为探究参数。子图划分时间的长短会影响划分出的子图的规模大小，即子图中包含的点、边数量会受到影响。迭代次数的设定会影响算法的运行时间，合理的迭代次数可以大幅缩短算法训练时间上的损耗。本实验中，子图划分时间长短参数的选择区间为（10s，300s），步长为 10s，迭代次数的选择区间为（1，1000），评价准则为准确率，实验结果如图 5 和图 6 所示。

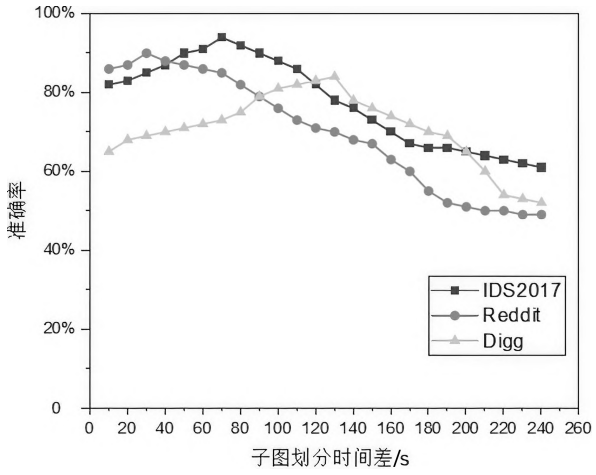

图 5 子图划分时间差对准确率的影响

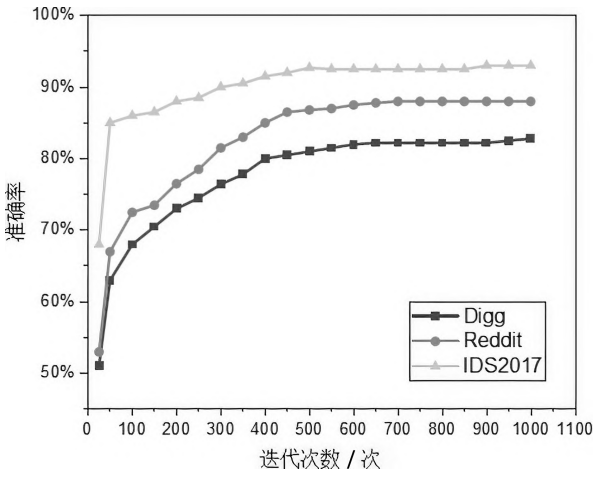

图 6 迭代次数对准确率的影响

图 5 展示了子图划分时间间隔对准确率的影响，从图中可以看出，对于 IDS2017、Reddit 和 Digg 这 3 种数据集，准确率出现峰值时对应不同的子图划分时间差，这是由于数据集中的数据特征导致的。其中 Reddit 在子图划分时间间隔为 30s 左右达到了峰值，这是由于 Reddit 数据集包含的点、边信息丰富，即使子图按 30s 的时间间隔作为划分，也拥有最优的点和边信息量，而 Digg 数据集包含的点、边信息较为疏松，导致其更适合较长的时间间隔对子图进行划分，最优的时间间隔为 120s 左右，而 IDS2017 的最优划分时间间隔为 60s 左右。对于不同数据集的相同点是，当子图划分的时间间隔小于最优值时，准确率会随着时间间隔的增加而增加，这是由于子图的点、边信息会随着时间间隔

的增加而逐渐丰富。当子图划分时间间隔大于最优值时，准确率会随着时间间隔的增加而降低，这是由于子图中的点、边信息过于丰富，导致出现了大量的冗余信息，对实验产生了噪声，不必要的点信息和边连接对实验产生干扰，导致准确率降低。

图6展示了迭代次数对准确率的影响，从图6中可以看出，随着迭代次数的增加，在IDS2017、Reddit和Digg数据集上检测的准确率总体呈上升趋势，在迭代次数为500左右时，逐渐收敛，继续增加迭代次数对准确率的提升并不明显，反而增加了一定时间成本，故选择迭代次数为500作为实验最佳的参数选择。

在选择最佳参数的基础上，将DGCNAE模型与其他算法进行对比试验，选用的对比方法有DeepWalk[23]、GraphSage[24]和Spotlight[25]。

本文提出的基于双重图神经网络和自编码器算法的网络异常检测方法DGCNAE和其他对比方法获得的实验准确率结果如表4所示。

表 4 不同模型实验结果对比

方法	IDS2017	Reddit	Digg
DeepWalk	70.3%	35.4%	69.8%
GraphSage	88.1%	82.2%	67.6%
Spotlight	86.4%	83.1%	75.3%
DGCNAE	**94.3%**	**90.6%**	**83.8%**

在表4中，采用准确率作为实验效果评价指标，加粗的实验结果为算法对比中最优结果，为了消除实验偶然性带来的影响，实验结果均为20轮重复实验的均值。为使实验结果更直观，实验准确率结果对比如图7所示。

实验结果表明，本文提出的DGCNAE方法与其他经典算法比较时，在准确率方面有较大的提升，这表明了DGCNAE在处理图数据时，不仅能对节点特征进行提取，还可以准确地捕捉到结构上的特征，这对于图流的异常检测具有重要作用。

对于DeepWalk方法，在3个数据集的实验结果中表现一般，在IDS2017数据集中获得最高准确率为70.3%，而在Reddit数据集上的实验准确率只有45.4%，这表明DeepWalk方法在面对图节点密集的图时存在一

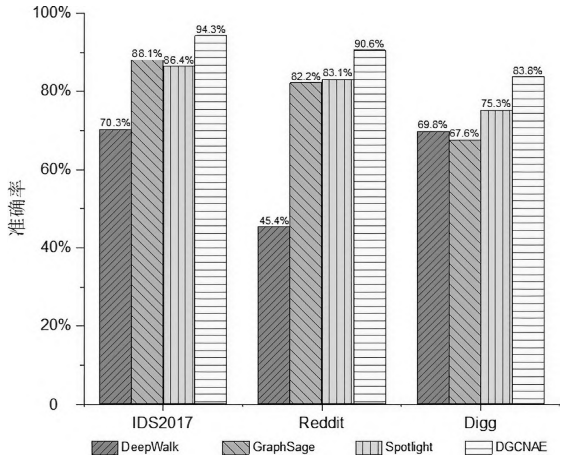

图 7 4 种算法准确率对比

定的缺陷，无法通过随机游走获取到准确的特征，这可能是由于DeepWalk的泛化能力不足导致的。

对于GraphSage方法，其效果略优于DeepWalk，IDS2017数据上的实验准确率为88.1%，而在Digg数据上获得结果与DGCNAE相差16.2%。该算法在IDS2017上的处理结果仅次于DGCNAE，表明该算法可以在一定程度上聚集节点周围的特征，当新的子图到来时，也能动态地对变化的特征进行学习，但是该方法和DeepWalk同样使用了skip-gram模型计算损失函数，这就导致学习得到的表示向量具有一定的经验误差。

对Spotlight算法，在3个数据集上的表现都不错，准确率都在80%以上，这表明该算法在应对图流的异常检测方面表现不错，但是与DGCNAE有一定的差距，这可能是由于在不同时刻到来的子图大小不统一，草图绘制时无法选取合适维度进行距离计算所导致。

对于本文提出的DGCNAE算法，在3个数据集上的实验准确率均高于其他3种算法，并且准确率均达到了90%以上。这表明算法在对图进行表示学习时，提取的特征最为全面，这是由于在进行特征提取过程中，不仅对节点进行了特征提取，还模仿节点特征的传播对边也进行了传播学习，使得最后获取的特征包含学习到的点和边的融合特征。

综上，通过与其他3种算法在3个不同的数据集上的实验对比，表明了DGCNAE具有更高的准确率和泛

化能力，在应对图的异常识别中，可以兼备属性和结构的特征，得到更准确的识别结果。

4 结束语

本文提出了一种基于双重图神经网络和自编码器的网络异常检测方法，该方法对通信数据进行图构建和子图划分。在子图级别上，通过引入两层图卷积神经网络分别对子图中的点和边进行特征提取，从而捕捉网络中局部和全局信息。该方法采用无监督学习方法对划分出的子图进行训练。通过迭代实验，本研究确定了最佳的子图划分时间间隔和迭代次数，以提高该方法的性能。实验结果表明，本文所提出的方法在多个数据集上均表现出较高的准确率和泛化能力，与其他算法相比具有明显的优势。

参考文献：

[1] YOST J R. The March of IDES: Early History of Intrusion-Detection Expert Systems[J]. IEEE Annals of the History of Computing, 2016, 38(4): 42-54.
[2] JAMES G, WITTEN D, HASTIE T, et al. An Introduction to Statistical Learning: With Applications in R[M]. New York: Springer, 2021.
[3] THAKKAR A, LOHIYA R. A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT: Recent Updates, Security Issues, and Challenges[J]. Archives of Computational Methods in Engineering, 2021, 28(4): 3211-3243.
[4] VINAYAKUMAR R, ALAZAB M, SOMAN K P, et al. Deep Learning Approach for Intelligent Intrusion Detection System[J]. IEEE Access, 2019, 7: 41525-41550.
[5] ESWARAN D, FALOUTSOS C. SedanSpot: Detecting Anomalies in Edge Streams[C]//IEEE. 2018 IEEE International Conference on Data Mining (ICDM). New York: IEEE, 2018: 953-958.
[6] ZHU Huidi, LU Jialiang. Graph-Based Intrusion Detection System Using General Behavior Learning[C]//IEEE. GLOBECOM 2022-2022 IEEE Global Communications Conference. New York: IEEE, 2022: 2621-2626.
[7] YAO Yepeng, SU Liya, LU Zhigang. DeepGFL: Deep Feature Learning via Graph for Attack Detection on Flow-Based Network Traffic[C]//IEEE. MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM). New York: IEEE, 2018: 579-584.
[8] FANG Yong, WANG Congshuang, FANG Zhiyang, et al. LMTracker: Lateral Movement Path Detection Based on Heterogeneous Graph Embedding[J]. Neurocomputing, 2022, 474: 37-47.
[9] WU Zonghua, PAN Shirui, CHEN Fengwen, et al. A Comprehensive Survey on Graph Neural Networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 32(1): 4-24.
[10] CAVILLE E, LO W W, LAYEGHY S, et al. Anomal-E: A Self-Supervised Network Intrusion Detection System Based on Graph Neural Networks[J]. Knowledge-Based Systems, 2022, 258: 1-12.
[11] KIPF T N, WELLING M. Semi-Supervised Classification with Graph Convolutional Networks[EB/OL]. (2016-09-09)[2023-05-13]. https://arxiv.org/abs/1609.02907.
[12] LIU Jie, LI Xiwang. Anomaly Detection Algorithm in Industrial Control Network Based on Graph Neural Network[J]. Computer Systems & Applications, 2020, 29(12): 234-238.
刘杰，李喜旺 . 基于图神经网络的工控网络异常检测算法 [J]. 计算机系统应用，2020，29（12）: 234-238.
[13] TANG Qiuhang, CHEN Huadong, GE Binbin, et al. AIGCN: Attack Intention Detection for Power System Using Graph Convolutional Networks[J]. Journal of Signal Processing Systems, 2022, 94(11): 1119-1127.
[14] POURHABIBI T, ONG K L, KAM B, et al. Fraud Detection: A Systematic Literature Review of Graph-Based Anomaly Detection Approaches[J]. Decision Support Systems, 2020, 133: 1-15.
[15] KIM H, LEE B S, SHIN W Y, et al. Graph Anomaly Detection with Graph Neural Networks: Current Status and Challenges[J]. IEEE Access, 2022, 10: 111820-111829.
[16] DING Qingfeng, LI Jinguo. A Distributed Abnormal Traffic Detection Scheme in the Internet of Things Environment[J]. Computer Engineering, 2022, 48(8): 152-159.
丁庆丰，李晋国 . 一种物联网环境下的分布式异常流量检测方案 [J]. 计算机工程，2022，48（8）: 152-159.
[17] ZOLA F, SEGUROLA-GIL L, BRUSE J L, et al. Network Traffic Analysis through Node Behaviour Classification: A Graph-Based Approach with Temporal Dissection and Data-Level Preprocessing[J]. Computers & Security, 2022, 115: 1-17.
[18] LO W W, LAYEGHY S, SARHAN M. E-GraphSAGE: A Graph Neural Network Based Intrusion Detection System for IoT[C]//IEEE/IFIP. 2022 IEEE/IFIP Network Operations and Management Symposium (NOMS 2022). New York: IEEE, 2022: 1-9.
[19] LIU Jinpeng. Network Security Protection Based on Machine Learning Technology[J]. Cybersecurity, 2018, 9(9): 96-102.
刘金鹏 . 基于机器学习技术的网络安全防护 [J]. 网络空间安全，2018，9（9）: 96-102.
[20] EDDAHMANI I, PHAM C H, NAPOLEON T, et al. Unsupervised Learning of Disentangled Representation via Auto-Encoding: A Survey[J]. Sensors, 2023, 23(4): 1-19.
[21] SAKURADA M, YAIRI T. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction[C]// MLSDA. Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis. New York: ACM, 2014: 4-11.
[22] NOBLE C C, COOK D J. Graph-Based Anomaly Detection[C]// ACM SIGKDD. The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2003: 631-636.
[23] PEROZZI B, AL-RFOU R, SKIENA S. DeepWalk: Online Learning of Social Representations[C]//ACM SIGKDD. The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2014: 701-710.
[24] HAMILTON W, YING R, LESKOVEC J, et al. Inductive Representation Learning on Large Graphs[J]. Neural Information Processing Systems, 2017, 1(2): 1024-1034.
[25] ESWARAN D, FALOUTSIS C, GUHA S, et al. SpotLight: Detecting Anomalies in Streaming Graphs[C]//ACM SIGKDD. KDD '18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM, 2018: 1378-1386.