基于并行流量图和图神经网络的加密流量分类方法

刘涛涛[1]，付钰[1]，俞艺涵[2]，安义帅[1]
（1.海军工程大学信息安全系，湖北 武汉 430033；2.海军工程大学作战运筹与规划系，湖北 武汉 430033）

摘　要：针对传统加密流量分类方法受限于数据集类不平衡以及复杂网络环境下所用特征不可靠等问题，提出一种基于并行流量图和图神经网络的加密流量分类方法。首先，从数据包头部和有效负载2个角度分别构建流量图以突出二者的差异；其次，引入改进的图注意力网络提取并行流量图的有效信息；然后，利用特征交叉融合注意力模块将提取到的信息进行融合以获得更为鲁棒的特征表示；最后，通过全连接层和Softmax层进行分类。实验表明，所提方法在ISCX-VPN、ISCX-nonVPN、ISCX-Tor和ISCX-nonTor数据集上取得了较好的效果，准确率分别为96.88%、90.62%、99.24%和98.13%，有效提升了加密流量分类性能。

关键词：加密流量分类；深度学习；图神经网络；特征融合
中图分类号：TP393
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025095

Encrypted traffic classification method based on parallel traffic graph and graph neural network

LIU Taotao[1], FU Yu[1], YU Yihan[2], AN Yishuai[1]
1. Department of Information Security, Naval University of Engineering, Wuhan 430033, China
2. Department of Operational Operations and Planning, Naval University of Engineering, Wuhan 430033, China

Abstract: Aiming at the problems of traditional encrypted traffic classification methods limited by the imbalance of dataset classes and the unreliability of the features used in complex network environments, an encrypted traffic classification method based on parallel traffic graph and graph neural network was proposed. Firstly, the traffic graphs were constructed from the packet header and payload perspectives to emphasize their differences. Then, an improved graph attention network was introduced to extract effective information from the parallel traffic graphs. Next, a feature cross-fusion attention module was used to fuse the extracted information, achieving a more robust feature representation. Finally, classification was performed using fully connected layers and a Softmax layer. Experiments show that the proposed method achieves better results on the ISCX-VPN, ISCX-nonVPN, ISCX-Tor, and ISCX-nonTor datasets, with accuracies of 96.88%, 90.62%, 99.24%, and 98.13%, respectively, significantly enhancing encrypted traffic classification performance.

Keywords: encrypted traffic classification, deep learning, graph neural network, feature fusion

0　引言

流量分类是网络安全领域的重要组成部分，在提升网络服务质量（QoS, quality of service）以及用户体验效果（QoE,quality of experience）等方面发挥着关键作用[1]。近年来，网络流量呈加密化趋势，在保护互联网用户隐私和满足匿名要求的同时也对流量分类带来了新的挑战，例如，攻击者为隐藏自身行为会通过加密技术来逃避检测，从而对系

收稿日期：2025–03–10；修回日期：2025–05–19
通信作者：俞艺涵，yuyihan333@163.com
基金项目：国家自然科学基金资助项目（No.62102422）；河南省科技攻关基金资助项目（No.242102211070）
Foundation Items：The National Natural Science Foundation of China (No.62102422), Henan Province Key Science and Technology Research Project of China (No.242102211070)

统进行入侵和攻击[2]。此外，一些隐私增强工具的使用让网络环境变得愈加复杂，如VPN（virtual private network）和Tor（the onion router），在一定程度上提升了流量分类的难度。因此，对加密流量进行分类已成为当前亟待解决的问题，引起了学术界和工业界的高度关注。

早期的网络流量分类方法主要依靠基于端口[3]和深度包检测（DPI, deep packet inspection）[4]。基于端口的方法利用端口与协议之间的对应关系进行分类，但随着未注册以及随机端口的激增，该方法变得无效。基于DPI的方法通过分析数据包识别特定的协议和应用数据，不过只适用于未加密的流量，在面对加密流量时存在一定的局限性，因此该方法也不再可靠。

为了应对加密化带来的挑战，学术界通过基于机器学习（ML, machine learning）的方法分析加密流量中的特征信息进行流量分类。该方法包括特征工程和模型训练2个阶段：前者是从网络流量中提取出统计特征，如数据包长度[5]、到达时间间隔[6]等；后者则是将统计特征输入分类器中进行训练，如随机森林（RF, random forest）[7]、决策树（DT, decision tree）[8]等。不过该方法依赖于人工提取特征，耗时较长，而且随着网络环境的动态发展，所提取的统计特征不再可靠且分类器也难以挖掘输入的深层特征，因此，亟须提出新的分类技术。

近年来，作为机器学习的一个分支，深度学习（DL, deep learning）引发了一场关于自主特征学习的研究热潮。该方法通过端到端的方式直接从原始流量中提取特征进行分类并取得了较好的效果，省去了人工特征提取的步骤，已成功应用在加密流量分类领域，如文献[9]和文献[10]提出的卷积神经网络（CNN, convolutional neural network）和长短期记忆（LSTM, long short-term memory）网络等模型。不过CNN和LSTM等模型在复杂多变的网络流量环境中特征提取能力有限，无法从加密流量中捕获有效的信息数据，所以仍需探索新方法。

当前，图神经网络（GNN, graph neural network）因其在处理非结构化数据上的强大能力逐渐得到广泛应用，其通过识别流量图中的拓扑结构进行分类。目前，大部分基于GNN的方法普遍基于数据包之间的相关性来构建图，然而这是统计特征的另一种使用形式，仍然存在前述问题。其他方法虽然通过数据包字节避免了该问题，却忽略了数据包头部（Header）和有效负载（Payload）之间的差异，只是简单地将二者作为整体进行学习，没能充分利用好二者所包含的信息。

此外，由于网络服务的普及程度不同，因此加密流量数据集中通常会出现类不平衡问题，即通常应用的样本数量要远超于其他应用的样本数。这种类不平衡问题会导致模型在分类过程中更倾向多样本数类别，忽略少数样本数类别或直接将其淹没，而分类器又是基于数据集分布平衡假设，所以在处理类不平衡数据集时其性能将会退化，这对于网络资源管理及网络安全都会产生影响。然而，当前加密流量分类方法中鲜有针对类不平衡问题的研究[11-12]。

因此，为解决现有方法中存在的特征不可靠、数据包头部和有效负载混合使用及数据集类不平衡等问题，本文提出一种基于并行流量图和图神经网络（PTG-GNN, parallel traffic graph and graph neural network）的加密流量分类方法。首先，基于原始字节之间的相关性，对预处理后的Header和Payload分别构建流量图作为模型的输入。同时利用改进的图注意力网络（GATv2, graph attention networks v2）进行特征提取并将流量图映射成高维图向量，有效捕获加密流量中隐含的深层次关联信息，克服传统特征工程中特征不可靠问题。其次，通过时间融合注意力模块（TFAM, temporal fusion attention module）对Header图向量和Payload图向量进行融合，形成一个综合向量。可充分发掘数据包头部和有效负载的不同语义信息，构建更具辨识能力的特征表示，避免因二者混合使用导致信息模糊的问题。最后，采用全连接层完成端到端训练以得到分类结果。此外，本文引入Equalization Loss v2（EQLv2）作为损失函数，通过平衡正负梯度比以解决数据集中普遍存在的类不平衡问题，增强模型对少数类样本的学习能力，从而提升整体分类性能。在实验部分，本文方法在ISCX VPN-nonVPN和ISCX Tor-nonTor这2个公开数据集上开展了一系列实验，旨在验证本文方法的有效性和合理性。

本文工作的主要贡献如下。

1)针对现有加密流量分类方法混合使用Header和Payload的问题，提出了一种基于字节间相关性的并行流量图，以突出两者之间的差异性，可为后

续加密流量分类提供更为高效的输入。

2)针对现有加密流量分类方法存在特征不可靠的问题，构建了一种PTG-GNN模型，利用GATv2分别提取Header和Payload流量图的特征信息，并通过TFAM进行融合以形成整体图向量表示，可有效提升加密流量细粒度分类性能。

3)针对加密流量数据集中的类不平衡问题，提出了一种具备梯度引导重加权机制的EQLv2作为损失函数，该函数根据每个分类器所累积的正负梯度比，分别对正梯度增加权重和负梯度降低权重，通过这种重加权机制有效提升了模型的分类性能，以较小的代价缓解了类不平衡问题。

4)通过在ISCX VPN-nonVPN和ISCX Tor-nonTor这2个具有挑战性的公开加密流量数据集上进行实验，验证了本文方法的有效性，同时实验性能也整体优于现有加密流量分类方法。

1　相关工作

本节将阐述近年来国内外学者在加密流量分类任务上提出的研究方法及最新进展。

1.1　基于机器学习的加密流量分类

传统机器学习方法主要依赖人工提取的特征，在过去的几十年里一直扮演着较为重要的角色且应用较为广泛，各种ML方法层出不穷并取得了良好的分类性能。

Taylor等[5]提出一种AppScanner方法用于加密流量分类，该方法通过从流数据中提取数据包长度和间隔时间等统计特征，然后利用随机森林对输入进行分类以识别移动应用。Shen等[8]提出一种网页指纹识别方法FineWP，该方法提取客户端和服务端双向交互过程中的数据包长度累积和块、序列特征、统计特征作为输入，然后通过RF、K最近邻（KNN， k-nearest neighbor）和DT等机器学习算法对上述特征进行处理，并创建网络指纹，从而实现细粒度识别。Zaki等[7]提出一种GRAIN的加密流量分类方法，通过将2个随机森林分类器链接在一起，对数据包有效载荷长度的7种统计特征进行处理，从而实现不同分类粒度的操作，该方法在ISCX VPN-nonVPN数据集上与4种基线分类器进行比较均保持较好的性能。Han等[13]则针对加密恶意流量提出一种轻量级无监督分类模型，首先将从数据包中提取的统计信息进行特征压缩以提高模型运行效率，然后采用经典的K-means聚类算法实现分类，解决了实际场景中难以获得丰富的高质量标签问题，最后在公开加密恶意软件流量数据集DataCon2020上实现了90%以上的准确率。 Koumar等[6]设计了一种新颖的流量特征扩展方法，并基于统计、时间、频率、分布和行为提出69个通用特征，然后在15个知名的公开数据集上使用RF、KNN及支持向量机（SVM, support vector machine）在内的14个传统机器学习算法进行验证；实验表明极限梯度提升 （XGBoost, extreme gradient boosting）算法的性能最佳，在ISCX-VPN、UNSW-NB15数据集上分别取得了94.35%和98.49%的分类准确率。不过上述方法都是基于特征工程，需要人工提取特征，对研究人员的知识储备要求较高，同时所耗费的时间较长且鲁棒性较差，因此，亟须寻求新型学习算法以跳出机器学习方法的局限性。

1.2　基于深度学习的加密流量分类

近年来，深度学习因其具有从海量数据中自动提取特征的能力，已成为加密流量分类领域中流行的解决方案，可有效提高分类的准确性。

Wang等[9]首次将端到端的深度学习技术引入加密流量分类领域，该方法使用一维CNN自动从原始流量数据中提取特征并学习其非线性关系，以更好地刻画加密流量的一维序列特性，通过在ISCX VPN-nonVPN数据集上的实验可知，该方法在性能上得到较大提升。Liu等[14]提出一种端到端模型FS-Net，通过双向门控循环单元去挖掘数据包长度序列中的时间信息，并且在编码-解码结构中加入重构机制以增强特征的有效性。Lin等[10]针对工业物联网中的海量流量提出一种结合CNN和LSTM的加密流量识别方案TSCRNN，该方案首先通过CNN提取抽象的空间特征，然后引入堆叠的双向LSTM学习时间特征，最后TSCRNN在ISCX Tor-nonTor上进行实验并取得了较高的准确率，因此仅使用少量数据包也可实现流量的早期识别。Lin等[15]基于Transformer设计了一种双向编码预训练模型ET-BERT，该方法首先在大量无标签的加密流量中学习通用的流量表示，然后在少量有标记数据上进行微调，从而完成特定任务。Liu等[2]设计了一种新的方案ATVITSC，首先将原始流量中的Session转换为灰度图像，保留了数据包之间的

顺序特性；然后并行馈送到分组视觉转换器和时空特征提取模块中，以捕获全局和时空特征；最后利用动态加强机制对并行特征进行融合，从而实现加密流量分类。虽然上述方法取得了不错的效果，但是从原始字节中提取特征的过程较为耗时，而且随着加密协议的更新迭代，大部分模型捕获关键性信息的能力也在下降。

1.3　基于图神经网络的加密流量分类

现实场景中加密流量为非结构化数据，而图神经网络在处理非结构化数据方面具备较大的潜力，已被广大学者应用到加密流量分类领域。

Shen等[16]提出指纹识别方法GraphApp，其基于客户端和服务器交互过程中产生的数据包构建了流量交互图，包含较为丰富的信息表示，然后利用强大的图神经网络进行分类，该方法在封闭场景和开放场景中都取得了不错的性能。Zhang等[17]基于逐点互信息（PMI, pointwise mutual information）构造了字节级流量图作为GNN的输入，捕获了原始字节之间隐含的相关性，实现了较好的效果。Han等[18]提出一种双重嵌入图神经网络模型DE-GNN，该方法首先分别对Header和Payload进行嵌入编码，然后通过CNN提取Header和Payload的包级别特征并以此构建流量交互图，最后通过GNN提取流级别特征并融合，从而实现加密流量分类。Li等[19]基于通信行为提出一种新的图结构，该图结构的节点为交互状态，边为转移状态，然后再通过GNN捕获子图进行分类。Cui等[20]利用原始流量中的数据包构建流图，具备丰富的节点属性，然后通过集成2类GNN提取特征以实现分类识别。不过上述方法都未能解决类不平衡问题，而DL又是在数据驱动下进行，需要大量的数据进行训练，因此少数类数据容易被多数类数据淹没，从而使得DL模型无法提取少数类数据的有效信息，所以类不平衡也是加密流量分类领域中亟待解决的问题。

综上所述，本文对现有加密流量分类方法进行了对比，具体如表1所示。由表1可知，GNN方法总体上表现较好，基本上能实现90%以上的准确率，而ML方法和DL方法虽然也有准确率较优的时候，但所消耗的计算成本较高，因此通过GNN来实现加密流量分类不失为一种可行的方法。

2　PTG-GNN模型架构构建

PTG-GNN模型框架如图1所示，包括并行流量图构建、双重嵌入、图特征提取与融合以及流量分类4个部分。

表1　现有方法对比

文献	输入	分类器	原始字节	数据集	结果（准确率）
文献[5]	数据包长度、间隔时间	RF	否	ISCX-VPN	88.89%
文献[8]	数据包长度、统计特征	RF, DT, KNN	否	YH datasets	92.40%
文献[7]	有效负载长度	RF	否	ISCX-VPN	81.29%
文献[13]	统计特征	K-means	否	DataCon2020	超过90.00%
文献[6]	统计特征	RF, SVM, KNN	否	ISCX-VPN	94.35%
文献[9]	原始数据包字节	CNN	是	ISCX-VPN	83.59%
文献[14]	数据包长度	AE, GRU	否	ISCX-VPN	92.98%
文献[10]	数据包字节	CNN, LSTM	是	ISCX-Tor	90.90%
文献[15]	数据包字节	BERT	是	ISCX-VPN	98.90%
文献[2]	数据包字节	Transformer, CNN, LSTM	是	ISCX-VPN	97.89%
文献[16]	数据包长度和方向	GNN	否	Private	89.22%
文献[17]	头部字节、有效负载字节	GNN, LSTM	是	ISCX-VPN	95.91%
文献[18]	头部字节、有效负载字节	GNN, CNN	是	ISCX-VPN	96.88%
文献[19]	序列信息	GNN	否	ISCX-VPN	94.52%
文献[20]	数据包字节	integrates GNN	是	CIC-IOT2023	95.16%

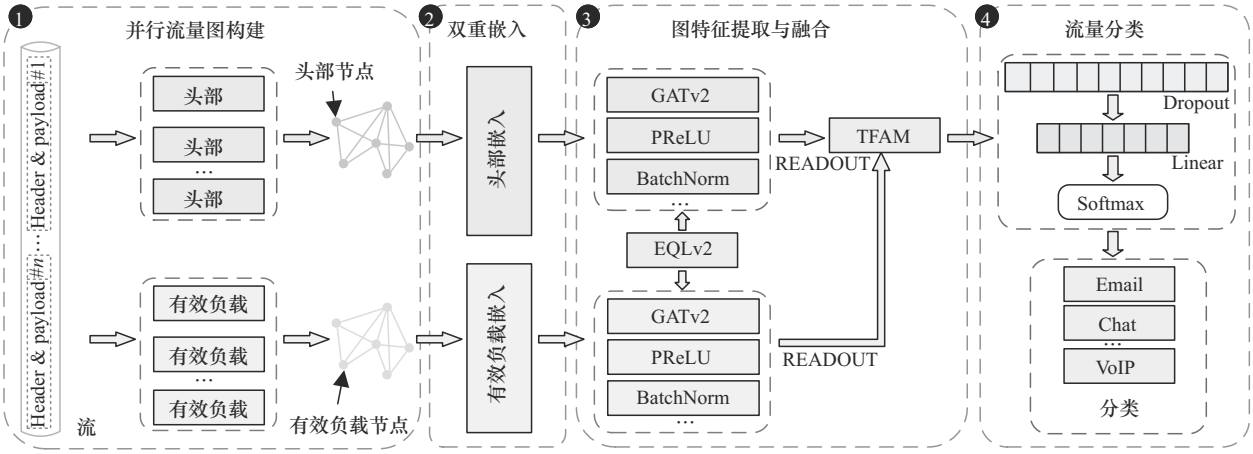


图1　PTG-GNN 模型架构

2.1　并行流量图构建

研究表明[17]，流量流中的统计特征在某些情况下显得尤为不稳定，从而导致加密流量分类性能下降。为此，本文将研究目光转向原始字节，这主要是因为原始字节作为基本的数据单元，其包含网络流量更为全面、具体的信息，通常被用来表示数据包。同时，字节的含义不仅与字节值有关，而且还取决于其所处的位置，也就是说，2 个相同字节值的字节在 Header 或 Payload 中的解释完全不同。其原因在于 Header 中主要为五元组信息（源 IP、源端口、目的 IP、目的端口及传输层协议），而 Payload 主要用来承载传输内容。如果将两者进行混合来构建流量图，会使得字节的含义模糊，从而导致模型难以收敛。综上所述，本文将从 Header 和 Payload 这 2 个角度分别构建流量图，即并行流量图。

通常来说，图一般被定义为 $G=\{V,E,X\}$，其中，V 表示节点的集合，E 为描述节点之间相连关系的边的集合，X 为节点初始特征的集合。为了更清晰地阐述并行流量图的构建过程，本文以包含 7 个字节的序列为例进行说明，并行流量图构建示意如图 2 所示。首先利用大小为 5 的滑动窗口提取该序列字节，其次统计窗口中各字节对的共现频率，然后基于逐点互信息来创建边，最后得到相应的图结构。

节点：字节是网络流量数据的最小单元，其语义由值和所在上下文共同决定，因此对字节粒度建立图结构可充分捕捉加密流量中的模式与结构信息。故本文将字节序列中的每一个字节都视为集合 V 中的一个节点，从而挖掘了字节之间的隐含关系。同时为了使本文方法更加简单高效，所有字节值相同的字节都共享同一个节点，即并行流量图中的节点不高于 256，提高了加密流量分类速度。

边：PMI 是自然语言处理中用来进行词关联计算的常用方法，而在网络流量分析中，PMI 能有效捕获字节之间的统计依赖关系，反映了特定应用或协议的通信模式，即使在加密环境下该模式仍会在字节分布和共现关系上表现出独特特征。因此，本文基于 PMI 来度量字节序列中 2 个字节之间的相关性[21]，并以此创建一条边。具体而言，本文将 2 个共现字节分别设置为 i_1 和 i_2，其中，共现字节是指在同一滑动窗口内同时出现的 2 个字节，然后依据式(1)来计算它们之间的 PMI 值。

$$\mathrm{PMI}(i_1,i_2)=\mathrm{lb}\frac{p(i_1,i_2)}{p(i_1)p(i_2)} \tag{1}$$

其中，$p(i_1,i_2)$ 表示 2 个字节同时出现的概率，$p(i_1)$

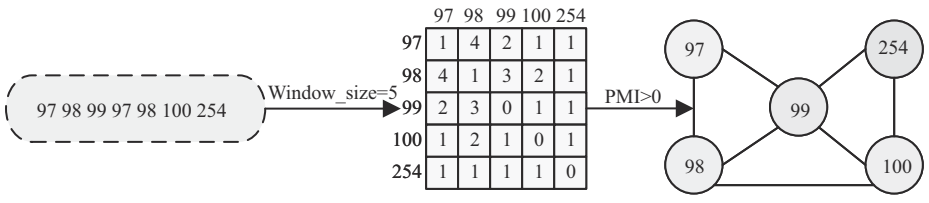


图2　并行流量图构建示意

和$p(i_2)$分别表示字节i_1和i_2出现的概率。值得说明的是，i_1和i_2只在PMI值为正值时建立一条边，并且邻接矩阵中的对应值为1；其他时候不建立边，邻接矩阵中的对应值为0，因为i_1和i_2此时几乎不存在相关性。

综上所述，本文通过上述步骤来构造并行流量图，同时由于$\mathrm{PMI}(i_1,i_2)=\mathrm{PMI}(i_2,i_1)$，因此本文所构造的并行流量图为无向图。

2.2 双重嵌入

由2.1节可知，本文将Header和Payload进行区分，构造了并行流量图，即流量Header图和流量Payload图，故将存在2个不同的嵌入向量。因此，本文通过构建2个嵌入层分别对流量Header图和流量Payload图进行处理，将字节值特征转换为高维向量，从而得到Header和Payload的有效嵌入。

具体来说，本文首先从并行流量图中选择需要嵌入的字节个数，分别记为N_H和N_P。然后将所选的字节嵌入为高维向量，同时由于一个字节的取值范围为[−128,127]，因此本文将维度d设置为256。最后流量Header图和流量Payload图的嵌入矩阵分别为$\boldsymbol{E}_H\in\mathbb{R}^{N_H\times 256}$和$\boldsymbol{E}_P\in\mathbb{R}^{N_P\times 256}$，其中每一个行向量表示一个字节。

2.3 图特征提取与融合

2.3.1 图特征提取

由前文可知，GNN在加密流量分类领域有着巨大的潜力，因此，本文将利用GNN对所获得的嵌入矩阵进行特征提取以更好地捕获流量信息。其中，图注意力网络（GAT, graph attention network）[22]作为一种空间域GNN已成为最为流行的架构之一。不过文献[23]认为GAT中的注意力机制为静态注意力机制，存在表达能力有限、无法拟合训练数据等问题。因此，为了克服GAT的局限性，本文将GATv2引入加密流量分类领域，该模型通过修改GAT的内部运算顺序形成动态注意力机制，以提升其表达能力，具体如下。

$$\mathrm{GAT}:e(\boldsymbol{h}_i,\boldsymbol{h}_j)=\mathrm{LeakyReLU}\left(\boldsymbol{a}^{\mathrm{T}}\left[\boldsymbol{W}\boldsymbol{h}_i\|\boldsymbol{W}\boldsymbol{h}_j\right]\right) \tag{2}$$

$$\mathrm{GATv2}:e(\boldsymbol{h}_i,\boldsymbol{h}_j)=\boldsymbol{a}^{\mathrm{T}}\,\mathrm{LeakyReLU}\left(\boldsymbol{W}\left[\boldsymbol{h}_i\|\boldsymbol{h}_j\right]\right) \tag{3}$$

其中，i和j为节点，$\boldsymbol{h}_i$和$\boldsymbol{h}_j$为节点向量，$e(\,)$为评分函数，用来评价邻居节点的特征对于本节点的重要性，$\boldsymbol{W}$为待学习的权重矩阵，$\boldsymbol{a}$为待学习的向量，$\boldsymbol{a}^{\mathrm{T}}$为$\boldsymbol{a}$的转置，$\|$为拼接操作，LeakyReLU是激活函数。从式(2)中可以看出，可学参数$\boldsymbol{a}$和$\boldsymbol{W}$是连续进行运算的，其可能分解成单个线性层，因此注意力系数的排序对图中任何一个节点都是相同的，即静态注意力机制。而GATv2将$\boldsymbol{a}$移到激活函数外，使查询键对先拼接再通过$\boldsymbol{W}$进行线性变换，避免了待学习参数连续应用的问题，提升了注意函数的表达能力，进而更加突出流量图中关键节点的特征。

综上可知，对于一个流量图而言，本文首先通过评分函数计算$e(\boldsymbol{h}_i,\boldsymbol{h}_j)$，其次再利用Softmax函数对节点进行归一化处理以确保各节点之间易于比较，具体如下。

$$\alpha_{ij}=\mathrm{Softmax}_j\left(e(\boldsymbol{h}_i,\boldsymbol{h}_j)\right)=\frac{\exp\left(e(\boldsymbol{h}_i,\boldsymbol{h}_j)\right)}{\sum\limits_{j'\in N_i}\exp\left(e(\boldsymbol{h}_i,\boldsymbol{h}_{j'})\right)} \tag{4}$$

其中，α_{ij}为归一化注意力系数，N_i为节点i所有邻居节点的集合，j'为节点i的任一邻居节点，exp( )为指数函数，Softmax则为激活函数。然后GATv2利用所得到的α_{ij}实现特征更新。

$$\boldsymbol{h}_i'=\sigma\left(\sum_{j\in N_i}\alpha_{ij}\boldsymbol{W}\boldsymbol{h}_j\right) \tag{5}$$

其中，$\boldsymbol{h}_i'$为节点i更新后的特征向量，σ为激活函数。同时本文将节点i在多个GATv2层的更新特征向量进行连接。

$$\boldsymbol{h}_i^{\mathrm{final}}=\mathrm{concat}\left(\boldsymbol{h}_i^1,\boldsymbol{h}_i^2,\cdots,\boldsymbol{h}_i^k\right) \tag{6}$$

其中，$\boldsymbol{h}_i^{\mathrm{final}}$为节点$i$最后的特征向量，$k$为本文模型所堆叠的GATv2层数。最后通过均值池化对流量图中所有的节点进行处理，从而得到最终的图特征向量。

$$\boldsymbol{g}=\frac{\boldsymbol{h}_1^{\mathrm{final}}\oplus\cdots\oplus\boldsymbol{h}_N^{\mathrm{final}}}{N} \tag{7}$$

其中，$\boldsymbol{g}$为最后的图特征向量，N为流量图中所有的节点数，⊕为加法运算。

2.3.2 TFAM

本文通过2.3.1节分别得到GATv2在Header和Payload上提取到的图特征向量。不过由于二者所蕴含的特征信息不同，因此本文考虑进行特征融合以提高分类准确率。但是传统的融合方法易受噪声干扰、鲁棒性不强，为此，文献[24]提出一种

TFAM方法利用通道和空间注意力来确定特征的重要部分，并通过跨空间维度的全局池化来聚合空间信息。因此，受文献[24]启发，本文采用TFAM对Header和Payload的最终图特征向量进行有效融合，TFAM结构示意如图3所示。

由图3可知，TFAM通过空间和通道2个维度来提取关键信息并进行融合。首先分别对Header向量和Payload向量进行平均池化和最大池化。

$$\begin{aligned}\boldsymbol{h}_a,\boldsymbol{p}_a &= \text{avgpool}\left(\boldsymbol{g}_h,\boldsymbol{g}_p\right)\\ \boldsymbol{h}_m,\boldsymbol{p}_m &= \text{maxpool}\left(\boldsymbol{g}_h,\boldsymbol{g}_p\right)\end{aligned} \tag{8}$$

其中，$\boldsymbol{h}_a,\boldsymbol{p}_a$和$\boldsymbol{h}_m,\boldsymbol{p}_m$分别表示Header向量和Payload向量经过平均池化和最大池化后的特征向量。将其进行拼接，有

$$\boldsymbol{s}_c,\boldsymbol{c}_c = \text{concat}\left(\boldsymbol{h}_a,\boldsymbol{h}_m,\boldsymbol{p}_a,\boldsymbol{p}_m\right) \tag{9}$$

其中，$\boldsymbol{s}_c$和$\boldsymbol{c}_c$分别表示空间维度和通道维度拼接后的特征向量，concat ( )为拼接操作。然后利用一维卷积确定二者的权重，并通过Softmax函数进行权重归一化。

$$\begin{aligned}\boldsymbol{W}_s &= \text{Softmax}\left(\text{conv}\left(\boldsymbol{s}_c\right)\right)\\ \boldsymbol{W}_c &= \text{Softmax}\left(\text{conv}\left(\boldsymbol{c}_c\right)\right)\end{aligned} \tag{10}$$

其中，$\boldsymbol{W}_s$和$\boldsymbol{W}_c$分别表示空间维度和通道维度的权重。最后，本文将上述权重与最初的Header向量和Payload向量进行相乘后再相加。

$$\boldsymbol{F}_V = \boldsymbol{W}_s\boldsymbol{g}_p + \boldsymbol{W}_c\boldsymbol{g}_h \tag{11}$$

其中，$\boldsymbol{F}_V$为Header向量和Payload向量的融合特征。通过上述步骤本文可以保留并行流量图中的关键信息，丢弃无用信息，从而实现并行流量图特征的有效融合。

2.4　加密流量分类

2.4.1　EQLv2

通常来说，加密流量分类方法使用交叉熵损失函数作为损失函数，定义为

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{L}\sum_{j=1}^{C} y_j^i \ln\left(p_j^i\right) \tag{12}$$

其中，$\mathcal{L}_{\text{CE}}$表示交叉熵损失函数，L为样本数，C为类别数，y_j^i表示如果样本i的真实类别为j取1，否则为0，p_j^i表示模型预测样本i属于类别j的概率。由式(12)可知，交叉熵函数平等对待每个样本，因此当数据集中存在类不平衡问题时，多数类样本会主导损失函数的优化方向，使模型在训练过程中忽视少数类样本的特征学习，从而导致模型在少数类样本上表现不佳，严重影响整体分类性能。为此，本文利用EQLv2作为损失函数以消除该问题。

模型输出针对损失函数正负梯度的计算过程为

$$\begin{aligned}\nabla_{z_j}^{\text{pos}}(\mathcal{L}) &= \frac{1}{|L|}\sum_{i\in L} y_j^i\left(p_j^i - 1\right)\\ \nabla_{z_j}^{\text{neg}}(\mathcal{L}) &= \frac{1}{|L|}\sum_{i\in L}\left(1 - y_j^i\right)p_j^i\end{aligned} \tag{13}$$

其中，z_j为模型输出，$\mathcal{L}$为损失函数。然后通过重加权系数对正负梯度进行处理，表达式为

$$q_j^{(t)} = 1 + \alpha\left(1 - f\left(g_j^{(t)}\right)\right), r_j^{(t)} = f\left(g_j^{(t)}\right) \tag{14}$$

其中，$q_j^{(t)}$和$r_j^{(t)}$为重加权系数，$g_j^{(t)}$为迭代t次以后所累计的比率，α为平衡系数，f ( )为映射函数，其表达式为

$$f(x) = \frac{1}{1 + \mathrm{e}^{-\gamma(x-\mu)}} \tag{15}$$

此时，正负梯度的更新方式为

$$\begin{aligned}\nabla_{z_j}^{\text{pos}'}(\mathcal{L}^{(t)}) &= q_j^{(t)}\nabla_{z_j}^{\text{pos}}(\mathcal{L}^{(t)})\\ \nabla_{z_j}^{\text{neg}'}(\mathcal{L}^{(t)}) &= r_j^{(t)}\nabla_{z_j}^{\text{neg}}(\mathcal{L}^{(t)})\end{aligned} \tag{16}$$

此外，还需对正负梯度下一次迭代的比率进行更新，表达式为

$$g_j^{(t+1)} = \frac{\sum_{t'=0}^{t}\left|\nabla_{z_j}^{\text{pos}'}(\mathcal{L}^{(t')})\right|}{\sum_{t'=0}^{t}\left|\nabla_{z_j}^{\text{neg}'}(\mathcal{L}^{(t')})\right|} \tag{17}$$

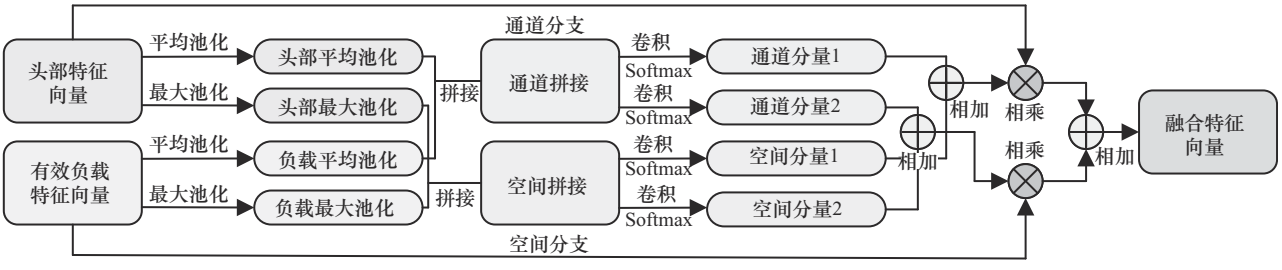


图3　TFAM结构示意

综上所述，本文通过EQLv2在训练阶段有效解决了类不平衡问题，避免了交叉熵损失函数的局限性，为加密流量分类提供了更为可靠的保障。

2.4.2 流量分类

通过上述部分得到Header流量图和Payload流量图的有效融合表示，以及通过EQLv2缓解类不平衡问题后，本文利用全连接层及Softmax层进行处理，从而得到最终的预测结果，最后通过EQLv2作为损失函数去计算损失并执行反向传播。

$$\text{loss} = \text{EQLv2}\left(\text{Softmax}\left(\text{FC}\left(\boldsymbol{F}_V\right)\right)\right) \tag{18}$$

其中，loss为损失值，FC为全连接层。

3 数据集与预处理

本节将对本文采用的加密流量数据集及其预处理流程进行详细的描述。

3.1 数据集

为了验证本文方法的有效性，将在ISCX VPN-nonVPN[25]及ISCX Tor-nonTor[26]这2个数据集上进行实验，具体标签类型如表2所示。

表2 ISCX VPN-nonVPN和ISCX Tor-nonTor数据集标签类型

数据集	标签	内容
ISCX-VPN	VPN-Email	Email、Gmail(SMTP、POP3、IMAP)
	VPN-Chat	ICQ、AIM、Skype、Facebook、Hangouts
	VPN-Streaming	Vimeo、Youtube、Netflix、Spotify
	VPN-File transfer	Skype、FTPS、SFTP
	VPN-VoIP	Facebook、Skype、Hangouts、Voipbuster
	VPN-P2P	uTorrent、Bittorrent
ISCX-nonVPN	Email	Email、Gmail(SMTP、POP3、IMAP)
	Chat	ICQ、AIM、Skype、Facebook、Hangouts
	Streaming	Vimeo、Youtube、Netflix、Spotify
	File transfer	Skype、FTPS、SFTP
	VoIP	Facebook、Skype、Hangouts、Voipbuster
	P2P	uTorrent、Bittorrent
ISCX-Tor	Tor-Browsing	Firefox、Chrome
	Tor-mail	Thunderbird(SMTP/S、POP3/SSL、IMAP/SSL)
	Tor-Chat	Facebook、Hangouts、Skype、AIM、ICQ
	Tor-Audio	Spotify
	Tor-Video	YouTube、Vimeo
	Tor-File	Skype、SFTP、FTPS
	Tor-VoIP	Facebook、Hangouts、Skype
	Tor-P2P	Bittorrent
ISCX-nonTor	Browsing	Firefox、Chrome
	Emial	Thunderbird(SMTP/S、POP3/SSL、IMAP/SSL)
	Chat	Facebook、Hangouts、Skype、AIM、ICQ
	Audio	Spotify
	Video	YouTube、Vimeo
	FTP	Skype、SFTP、FTPS
	VoIP	Facebook、Hangouts、Skype
	P2P	Bittorrent

ISCX VPN-nonVPN 数据集包含 ISCX-VPN 和 ISCX-nonVPN 这 2 个数据集，含有 Email、Chat、P2P 等 7 类良性样本，以及 VPN-Email、VPN-Chat、VPN-P2P 等 7 类协议封装样本。不过由于标签混淆，本文仅使用 6 类良性样本和 6 类协议封装样本进行实验。

ISCX Tor-nonTor 包含 ISCX-Tor 和 ISCX-nonTor 这 2 个数据集，含有 Browsing、FTP、VoIP 在内的 8 类常规流量，以及 Tor-Browsing、Tor-FTP、Tor-VoIP 等 8 类 Tor 流量。

3.2　数据预处理

本文所采用的加密流量数据集中的文件皆为 Pcap 格式的原始流量文件，因此有必要对其进行预处理以符合模型所要求的格式。具体步骤如下：1)Pcap 文件分割，本文使用工具 Splitcap 对数据集中的文件进行拆分以获得双向流；2)数据清洗，本文对所拆分文件中的坏包或重传包进行丢弃，同时对于数据包长度为 0 的空流及长度大于 10 000 超长流也进行去除，主要是因为空流中不存在构成流量图的有效信息，而超长流中又存在大量坏包或重传包的干扰。此外，针对筛选后的数据包，本文将其所包含的不相干信息（如以太网头）及噪声信息（如源 IP 地址、目的 IP 地址）进行了去除。

4　实验设置

4.1　实验环境

本文在具有 64 位 Ubuntu 操作系统、Core i9-9900K CPU @ 3.60 GHz 16 处理器和 NVIDIA GeForce RTX 3090 GPU 的电脑上进行实验。此外，本文通过十折交叉验证进行一系列对比实验，从而确定模型中存在的一些关键参数。首先，本文将 Header 和 Payload 的最大长度分别设置为 40 和 150，PMI 窗口大小为 5。然后，在训练阶段使用 Adam 优化器，学习率为 1×10^{-3}，epoch 为 50，batchsize 为 32，Dropout 为 0.5，GATv2 层数为 4 层，注意力机制中的多头数为 2，EQLv2 函数中的平衡系数 α 设置为 4，映射函数 $f()$ 中的 γ 和 μ 分别设置为 12 和 0.8。

4.2　评价指标

为了评估本文方法的性能，本文选取准确率（Acc）、精确率（Pre）、召回率（Rec）及 F1 分数（F1）4 个指标进行实验，计算式如下。

$$\mathrm{Acc}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}} \tag{19}$$

$$\mathrm{Pre}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{20}$$

$$\mathrm{Rec}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{21}$$

$$\mathrm{F1}=\frac{2\times\mathrm{Pre}\times\mathrm{Rec}}{\mathrm{Pre}+\mathrm{Rec}} \tag{22}$$

其中，TP、TN、FP、FN 分别表示真阳性、真阴性、假阳性及假阴性，即正样本预测正确的数量、负样本预测正确的数量、正样本预测错误的数量、负样本预测错误的数量。

4.3　ISCX 数据集的实验结果

为了验证本文方法的可行性，本节在 ISCX 的 4 个数据集上开展实验，ISCX 数据集的实验结果如图 4 所示。

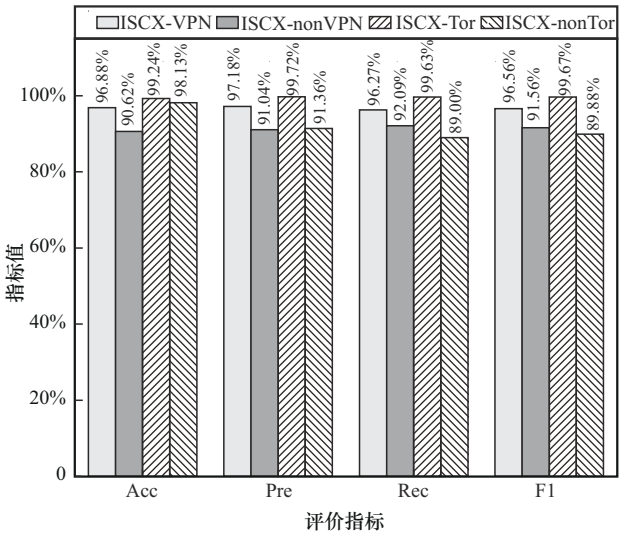


图 4　ISCX 数据集的实验结果

由图 4 可知，本文方法在 ISCX 的 4 个数据集上取得了较好的性能，基本能达到较高的结果。其中本文方法在 ISCX-Tor 数据集上的性能最好，4 个指标值都接近 100%。其次是在 ISCX-VPN 数据集上，所有指标值均在 96% 以上，较为稳定。本文在 ISCX-nonVPN 数据集上的结果也皆在 90% 以上，4 个指标值相差不大，具有不错的泛化性能。本文方法虽然在 ISCX-nonTor 数据集上的性能不如其他几个数据集，但这主要是因为该数据集中的类不平衡问题较为严重，不过本文在此情况下仍能取得 98.13% 的准确率，其他指标也能保持在 90% 左右。综上所述，本文方法即使在类不平衡的情况下也具

有较强的识别混淆流量的能力，有效验证了本文方法的可行性。

4.4　模型超参数分析

为探究模型中超参数对实验性能的影响，本文选取GATv2层数、注意力机制中的多头数、平衡系数α、γ及μ这5个超参数在ISCX-VPN数据集上进行实验，并选取准确率和F1分数作为评价指标，超参数分析结果如图5所示。

由图5可知，首先，随着GATv2层数逐渐增加，准确率和F1分数最初呈上升趋势，当层数为4时实验结果达到最优，此时继续增加层数将导致性能急剧下降，F1分数更是下降了大约20%；其次，本文将注意力机制中的多头数分别设置(1，2，3，4)，当多头数设置为2时准确率和F1分数的结果最优；最后，对于EQLv2损失函数中的3个超参数α、γ和μ，本文分别将其参数范围设置为(2，3，4，5，6)、(10，11，12，13，14)和(0.2，0.4，0.6，0.8，1.0)。从图5(c)~图5(e)可以看出，准确率和F1分数均呈先上升再下降趋势，当α、γ和μ分别为4、12和0.8时，评价指标达到最优。

4.5　与现有方法进行对比

为了验证本文方法的有效性，选取3种传统机器学习方法、7种深度学习方法及5种图神经网络方法作为基准模型与本文方法进行对比。

1)AppScanner[5]。AppScanner首先提取流数据中的统计特征，然后通过随机森林对其进行处理。

2)FlowPrint[27]。FlowPrint是一种半监督方法，通过相关性生成应用指纹并利用聚类进行识别匹配。

3)GRAIN[7]。GRAIN利用链接在一起的随机森林对Payload长度特征进行分类。

4)DF[28]。DF是一种基于CNN的深度指纹识别方法，只需要简单地输入格式便可实现较好的分类效果。

5)FS-Net[14]。Fs-Net使用双向GRU去挖掘数据包长度序列中的时间信息。

6)App-Net[29]。App-Net利用LSTM和CNN分别学习数据包长度和Payload序列特征。

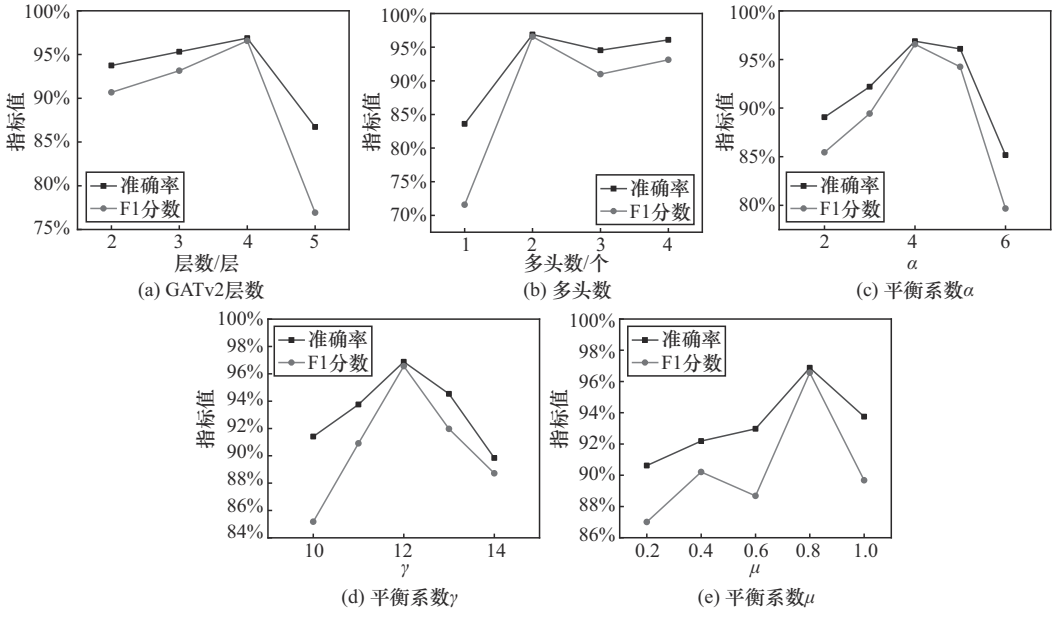
7)TSCRNN[10]。TSCRNN将CNN和双向LSTM进行结合，从而实现工业物联网加密流量分类。

8)FFB[30]。FFB使用CNN和RNN分别对原始字节和数据包长度序列进行学习。

9)ET-BERT[15]。ET-BERT从大量无标签数据中预训练出数据报级的表示，然后根据特定的任务再进行微调。

10)YaTC[31]。YaTC基于原始数据包字节创建多级流表示矩阵，然后利用新型流量Transformer模块进行学习。

11)GraphDApp[16]。GraphDApp利用数据包长度来构建流量交互图，然后利用图神经网络进行学习。



(a) GATv2层数　(b) 多头数　(c) 平衡系数α　(d) 平衡系数γ　(e) 平衡系数μ

图5　超参数分析结果

12)FB-GNN[32]。FB-GNN 通过数据包的时间顺序来构造图，并通过几何学习模型学习图特征。

13)TFE-GNN[17]。TFE-GNN 构建了一种字节级流量图，然后通过图卷积神经网络进行流量分类。

14)DE-GNN[18]。DE-GNN 利用突发包之间的关系构建了流量交互图，并通过图注意力网络进行特征提取。

15)MH-Net[33]。MH-Net 将不同数量的流量聚合成多视图流量图，然后利用异构图神经网络进行特征提取。

表 3 展示了本文方法与现有方法在 ISCX 数据集上的对比结果。由表 3 可以得到以下结论。 1)本文模型 PTG-GNN 在 4 个加密流量数据集上都呈现了较好的性能。首先在 ISCX-VPN 数据集上相较于其他方法，该模型的 4 个评价指标均排在前列，其中精确率更是提升了 3%~40%。而在 ISCX-nonVPN 数据集上，4 个指标虽然没能实现最优的分类性能，但其与最优指标值相差不大，基本只相差 0.2%~2%，且优于大多数基准模型。同时本文方法在 ISCX-Tor 和 ISCX-nonTor 这 2 个数据集上与其他基准模型相比，PTG-GNN 的 4 个指标基本都取得了最好的结果，并且都有着大幅度的提升，验证了本书方法的有效性。 2)AppScanner、FlowPrint 和 GRAIN 都没有取得令人满意的效果。在 ISCX 的 4 个数据集上，3 个传统的机器学习方法在 4 个指标值上普遍比 PTG-GNN 少大约 15%。这主要是因为传统的机器学习方法依赖人工提取特征，这一过程不可避免地存在信息的损耗，而且随着加密流量环境的愈加复杂，所提取到的特征也不再可靠有效，从而导致性能不理想。3)与 7 种深度学习方法相比较，PTG-GNN 在 4 个数据集上的性能要全面优于 DF、FS-Net、FFB 以及 App-Net 这 4 个模型，最低都取得了 10% 的提升。同时也只是在 ISCX VPN-nonVPN 和 ISCX-nonTor 数据集上存在个别指标值低于 TSCRNN、ET-BERT 和 YaTC。不过上述模型较为庞大复杂，消耗了大量的计算资源，而且在整体性能上明显低于 PTG-GNN。4)与 5 种图神经网络方法相比，本文方法的实验结果也令人满意。即便仍然存在某个数据集上的部分指标不如其他方法，但是差距并不大，而且在其他数据集上本文所取得的性能要远高于这些方法。GraphDApp 性能较差的原因主要是 VPN、Tor 等技术对数据包长度的干扰较大，导致所构造的图存在大量噪声。而 FB-GNN 的图是基于时间序列构造的，这意味着相同节点数的图所对应的图结构相同，因此方法鲁棒性不足，性能不太稳定。TFE-GNN 无法处理数据包原始字节中的隐含噪声，故模型结果稍差。DE-GNN 则在传递图内信息方面存在局限性，因此性能低于本文方法。MH-Net 未能考虑到类不平衡问题，从而对某些少数类样本的学习不够充分。 5)TFE-GNN 和 DE-GNN 也是基于 Header 和 Payload 这 2 个角度构建图，并且在表 3 中 2 种方法所取得的实验结果要优于其他方法，所以从 Header 和 Payload 这 2 方面对加密流量进行分类是可行、有效的。此外，大部分基准模型在不同数据集上的性能表现出较大差异，尤其是在 ISCX Tor-nonTor 数据集上，这是因为 Tor 数据集中的部分类别的样本有限。而本文方法通过 EQLv2 损失函数有效解决了该问题，因此在 4 个数据集都取得了较为优异、稳定的实验结果，泛化性较强。

4.6　与经典图神经网络进行对比

为了验证 GATv2 模型的可靠性，本文将其与 3 个经典的图神经网络架构（GCN、GAT、GraphSAGE）在 ISCX 数据集上开展实验并进行分析对比，不同图神经网络实验结果如表 4 所示。

由表 4 可知，本文所使用的 GATv2 在 4 个数据集上都取得了最好的结果。与 GraphSAGE 模型相比，GATv2 在大部分指标上提升了 4% 以上，尤其是在 ISCX-nonVPN 数据集的准确率指标上，GATv2 实现了 10.79% 的性能提升。这主要是因为 GraphSAGE 所考虑的邻居节点是固定的，因此在进行聚合操作时存在图信息丢失，并且同一节点的嵌入特征不够稳定。而在 ISCX-nonTer 数据集与 GCN 和 GAT 比较时，GATv2 也取得了一定程度上的性能提升，在兼顾精确率和召回率的综合指标 F1 分数上分别最大提升了 4.13% 和 2.42%。这是因为 GCN 假设图的拓扑结构是不变的，而在实际的加密流量环境中，其拓扑结构通常是动态变化的。GAT 虽然可以评估相邻节点的重要性，并且能在特征传播过程中利用这些信息，但是其使用的静态注意力机制存在表达能力弱、训练数据拟合差等缺点。GATv2 通过动态注意力机制不仅能克服这些缺点，而且鲁棒性也要更强。

表3 本文方法与现有方法在ISCX数据集上的对比结果

方法	ISCX-VPN				ISCX-nonVPN				ISCX-Tor				ISCX-nonTor			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
AppScanner	88.89%	86.79%	88.15%	87.22%	75.76%	75.94%	74.65%	74.86%	75.43%	66.29%	60.42%	61.63%	91.53%	84.35%	81.40%	82.73%
FlowPrint	85.38%	74.51%	79.17%	75.66%	69.44%	70.73%	73.10%	71.31%	24.00%	3.00%	12.50%	4.84%	52.43%	75.90%	60.74%	61.53%
GRAIN	81.29%	80.77%	81.09%	80.27%	66.67%	65.32%	66.64%	65.67%	69.14%	52.53%	53.46%	52.34%	78.95%	67.14%	66.15%	66.13%
DF	80.12%	77.99%	81.52%	79.21%	67.42%	68.57%	67.17%	67.01%	65.14%	48.03%	47.67%	47.19%	85.68%	80.03%	74.15%	75.90%
FS-Net	92.98%	92.63%	92.11%	92.34%	76.26%	76.85%	75.34%	75.55%	82.86%	74.87%	71.91%	72.42%	92.78%	83.68%	82.54%	82.85%
App-Net	96.09%	94.64%	95.73%	94.95%	87.11%	87.79%	87.60%	87.60%	71.43%	51.92%	53.74%	51.68%	—	—	—	—
TSCRNN	—	96.57%	**96.45%**	96.52%	—	88.87%	88.77%	88.73%	—	98.58%	98.49%	98.54%	—	91.23%	91.15%	91.10%
FFB	83.04%	87.14%	81.49%	83.35%	70.20%	72.74%	69.45%	70.50%	63.43%	48.70%	52.03%	49.52%	89.54%	75.45%	74.30%	74.30%
ET-BERT	95.32%	94.36%	95.07%	94.63%	**91.67%**	92.45%	**92.29%**	92.35%	95.43%	92.42%	96.06%	93.97%	90.29%	85.60%	82.17%	83.32%
YaTC	96.05%	—	—	96.71%	75.46%	—	—	75.44%	98.68%	—	—	98.69%	95.79%	—	—	**95.22%**
GraphDApp	64.91%	56.68%	61.03%	57.40%	44.95%	42.30%	36.47%	36.14%	42.86%	25.57%	25.09%	22.81%	69.36%	54.47%	53.98%	53.52%
FB-GNN	84.31%	82.93%	81.41%	81.84%	65.03%	65.58%	65.41%	63.91%	42.86%	23.81%	31.36%	25.81%	—	—	—	—
TFE-GNN	95.91%	95.26%	95.93%	95.36%	90.40%	**93.16%**	91.90%	**92.40%**	98.86%	97.92%	99.39%	98.55%	93.90%	87.42%	83.35%	85.07%
DE-GNN	96.88%	96.59%	95.94%	96.24%	89.84%	91.10%	90.07%	90.48%	98.72%	99.43%	99.39%	99.05%	—	—	—	—
MH-Net	**97.68%**	—	—	**97.66%**	88.22%	—	—	88.14%	99.16%	—	—	99.17%	91.56%	—	—	91.22%
PTG-GNN	96.88%	**97.18%**	96.27%	96.56%	90.62%	91.04%	92.09%	91.56%	**99.24%**	**99.72%**	**99.63%**	**99.67%**	**98.13%**	**91.36%**	**89.00%**	89.88%

表 4　　不同图神经网络实验结果

模型	ISCX-VPN				ISCX-nonVPN				ISCX-Tor				ISCX-nonTor			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
GCN	95.31%	95.07%	95.52%	95.04%	86.08%	88.45%	88.06%	88.10%	99.09%	99.38%	99.21%	99.27%	98.13%	87.82%	84.57%	85.75%
GAT	95.31%	93.65%	93.07%	93.15%	88.64%	88.85%	90.78%	89.62%	98.48%	97.84%	97.38%	97.59%	98.32%	88.67%	86.78%	87.46%
GraphSAGE	91.41%	91.30%	90.72%	90.86%	79.83%	80.56%	83.32%	80.36%	96.97%	98.61%	95.43%	96.77%	97.34%	87.27%	84.15%	84.83%
GATv2	**96.88%**	**97.18%**	**96.27%**	**96.26%**	**90.62%**	**91.04%**	**92.09%**	**91.56%**	**99.24%**	**99.72%**	**99.63%**	**99.67%**	**98.13%**	**91.36%**	**89.00%**	**89.88%**

4.7　模型复杂度对比

为了体现本文模型可在性能和复杂度之间取得权衡，本节从基准模型中挑选 2 种经典 CNN 方法（FS-Net、App-Net），3 种 GNN 方法（GraphDApp、FB-GNN、TFE-GNN），以及当前较为流行的 Transformer 方法 ET-BERT 与本文方法在模型大小上进行对比。对比方法具备一定的代表性，能够覆盖从轻量级网络到复杂结构网络的不同类型，确保了对比的全面性，同时也反映了加密流量分类方法的发展趋势，可充分验证本文模型的有效性，模型参数量对比如表 5 所示。

表 5　　模型参数量对比

模型	参数量
FS-Net	3.2×10^6
FFB	1.7×10^6
GraphDApp	1.1×10^4
TFE-GNN	4.4×10^7
App-Net	2.1×10^6
ET-BERT	8.6×10^7
FB-GNN	9.5×10^6
PTG-GNN	1.4×10^7

由表 5 和表 3 可知，PTG-GNN 在 ISCX 的 4 个数据集上取得显著性能的同时，模型复杂度也相对较小。与 PTG-GNN 相比，虽然 FS-Net、App-Net、FFB、GraphDApp 以及 FB-GNN 的复杂度不高，但是上述模型在 4 个数据集上的所取得的性能较差，无法有效识别出加密流量。此外，ET-BERT 和 TFE-GNN 即使在部分指标上优于 PTG-GNN，但是其参数量较大，其中 TFE-GNN 的参数量是 PTG-GNN 的 3 倍以上，而 ET-BERT 更是达到了 6 倍以上，这说明它们需要耗费更多的计算资源，推理时间也更长。本文 PTG-GNN 在实现更优异性能的同时还能节省计算成本，加快推理速度，在性能和复杂度之间实现了较好的权衡。

4.8　消融实验

为了验证本文方法中每个部分的必要性，本文将在 ISCX 的 4 个数据集上开展消融实验。同时为了便于展示，本文将 Header、Payload、Dual Embedding、TFAM 及 EQLv2 分别简写为 H、P、D、T 和 E，ISCX 数据集上的消融实验结果如表 6 所示。

由表 6 可知，Dual Embedding 层对实验结果的影响最大，4 个数据集中的指标值基本不超过 30%，这主要是因为 Header 和 Payload 中的数值为离散值，缺乏特定的含义和连续性，因此直接利用其进行分类无法取得令人满意的效果。同时由 w/o H 和 w/o P 实验可知，与 Payload 相比，Header 在加密流量分类工作中发挥的作用更大，并且不同数据集

表 6　　ISCX 数据集上的消融实验结果

模型	ISCX-VPN				ISCX-nonVPN				ISCX-Tor				ISCX-nonTor			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
w/o H	79.69%	81.41%	78.98%	78.29%	72.73%	73.18%	71.94%	72.02%	75.76%	82.05%	81.59%	77.04%	95.90%	82.16%	66.11%	69.92%
w/o P	92.97%	94.85%	88.05%	90.87%	86.08%	87.62%	87.60%	87.49%	96.97%	98.68%	91.00%	93.68%	96.60%	83.30%	81.00%	81.48%
w/o D	28.91%	35.41%	26.99%	20.21%	20.74%	21.78%	23.68%	15.25%	29.55%	20.51%	15.38%	11.83%	20.57%	18.73%	19.27%	8.67%
w/o T	94.53%	95.39%	94.02%	94.53%	88.92%	88.59%	90.88%	89.28%	96.97%	98.59%	89.59%	92.60%	97.67%	85.14%	85.53%	84.65%
w/o E	89.06%	86.71%	87.46%	85.83%	87.22%	87.03%	89.49%	87.78%	96.97%	95.77%	92.40%	92.77%	97.43%	81.91%	83.52%	82.26%
PTG-GNN	**96.88%**	**97.18%**	**96.27%**	**96.26%**	**90.62%**	**91.04%**	**92.09%**	**91.56%**	**99.24%**	**99.72%**	**99.63%**	**99.67%**	**98.13%**	**91.36%**	**89.00%**	**89.88%**

对于Header和Payload的依赖程度也不同。例如当Payload换成Header时，准确率在ISCX-nonTor数据集只提升了0.7%，而在ISCX-Tor数据集上却提升了21.21%。此外，TFAM的应用被证明可有效提升模型的性能，在ISCX-Tor数据集上对召回率实现提高了10.04%。最后由w/o E实验可以看出，EQLv2函数有效解决了加密流量数据集中的类不平衡问题，在4个数据集上对4个指标均实现了较大的提高。综上所述，消融实验充分证明了本文方法中每个部分都是必不可少的，对模型性能均取得了较大程度上的提升。

5 结束语

本文提出一种基于并行流量图和GNN的加密流量分类方法。不仅通过构建Header和Payload这2个流量图挖掘了原始字节之间隐含的相关性，而且还利用GATv2有效提取了并行流量图的内在特征信息并进行融合。同时还通过EQLv2较好地解决了加密流量数据集中的类不平衡问题。本文方法在ISCX VPN-nonVPN及ISCX Tor-nonTor数据集上开展仿真实验，并从准确率、精确率、召回率和F1分数4个指标对方法性能进行量化，在与多个基准模型进行对比时验证了PTG-GNN的有效性。不过本文忽略了字节序列中的时间信息，未来将考虑利用该信息以提升模型性能，同时在开放世界假设下开展实验也是下一步的研究重点。

参考文献：

[1] ZHAO J J, LI Q, HONG Y P, et al. MetaRockETC: adaptive encrypted traffic classification in complex network environments via time series analysis and meta-learning[J]. IEEE Transactions on Network and Service Management, 2024, 21(2): 2460-2476.

[2] LIU Y, WANG X, QU B, et al. ATVITSC: A novel encrypted traffic classification method based on deep learning[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 9374-9389.

[3] ERMAN J, MAHANTI A, ARLITT M, et al. Identifying and discriminating between web and peer-to-peer traffic in the network core[C]//Proceedings of the 16th International Conference on World Wide Web. New York: ACM Press, 2007: 883-892.

[4] YAN H N, LI H, XIAO M C, et al. PGSM-DPI: precisely guided signature matching of deep packet inspection for traffic analysis[C]//Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM). Piscataway:IEEE Press, 2019: 1-6.

[5] TAYLOR V F, SPOLAOR R, CONTI M, et al. AppScanner: automatic fingerprinting of smartphone apps from encrypted network traffic[C]// Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P). Piscataway: IEEE Press, 2016: 439-454.

[6] KOUMAR J, HYNEK K, ČEJKA T. Network traffic classification based on single flow time series analysis[C]//Proceedings of the 2023 19th International Conference on Network and Service Management (CNSM). Piscataway: IEEE Press, 2023: 1-7.

[7] ZAKI F, AFIFI F, ABD RAZAK S, et al. GRAIN: Granular multi-label encrypted traffic classification using classifier chain[J]. Computer Networks, 2022, 213: 109084.

[8] SHEN M, LIU Y T, ZHU L H, et al. Fine-grained webpage fingerprinting using only packet length information of encrypted traffic[J]. IEEE Transactions on Information Forensics and Security, 2020, 16: 2046-2059.

[9] WANG W, ZHU M, WANG J L, et al. End-to-end encrypted traffic classification with one-dimensional convolution neural networks[C]//Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). Piscataway: IEEE Press, 2017: 43-48.

[10] LIN K D, XU X L, GAO H H. TSCRNN: A novel classification scheme of encrypted traffic based on flow spatiotemporal features for efficient management of IIoT[J]. Computer Networks, 2021, 190: 107974.

[11] WANG P, LI S H, YE F, et al. PacketCGAN: exploratory study of class imbalance for encrypted traffic classification using CGAN[C]//Proceedings of the ICC 2020 - 2020 IEEE International Conference on Communications (ICC). Piscataway: IEEE Press, 2020: 1-7.

[12] VU L, VAN TRA D, NGUYEN Q U. Learning from imbalanced data for encrypted traffic identification problem[C]//Proceedings of the Seventh Symposium on Information and Communication Technology. New York: ACM Press, 2016: 147-152.

[13] HAN S B, WU Q H, ZHANG H, et al. Light-weight unsupervised anomaly detection for encrypted malware traffic[C]//Proceedings of the 2022 7th IEEE International Conference on Data Science in Cyberspace (DSC). Piscataway: IEEE Press, 2022: 206-213.

[14] LIU C, HE L T, XIONG G, et al. FS-net: a flow sequence network for encrypted traffic classification[C]//Proceedings of the IEEE INFOCOM 2019 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2019: 1171-1179.

[15] LIN X J, XIONG G, GOU G P, et al. ET-BERT: a contextualized datagram representation with pre-training transformers for encrypted traffic classification[C]//Proceedings of the ACM Web Conference 2022. New York: ACM Press, 2022: 633-642.

[16] SHEN M, ZHANG J P, ZHU L H, et al. Accurate decentralized application identification via encrypted traffic analysis using graph neural networks[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 2367-2380.

[17] ZHANG H Z, YU L, XIAO X, et al. TFE-GNN: a temporal fusion encoder using graph neural networks for fine-grained encrypted traffic classification[C]//Proceedings of the ACM Web Conference 2023. New York: ACM Press, 2023: 2066-2075.

[18] HAN X B, XU G Z, ZHANG M, et al. DE-GNN: Dual embedding with graph neural network for fine-grained encrypted traffic classification[J]. Computer Networks, 2024, 245: 110372.

[19] LI Y, CHEN X S, TANG W Y, et al. Interaction matters: Encrypted traffic classification via status-based interactive behavior graph[J]. Applied Soft Computing, 2024, 155: 111423.

[20] CUI S S, HAN X Y, HAN D Q, et al. FG-SAT: efficient flow graph for

encrypted traffic classification under environment shifts[J]. arXiv Preprint, arXiv: 2408.14122, 2024.

[21] YAO L, MAO C S, LUO Y. Graph convolutional networks for text classification[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33(1): 7370-7377.

[22] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks[J]. arXiv Preprint, arXiv: 1710.10903, 2017.

[23] BRODY S, ALON U, YAHAV E. How attentive are graph attention networks? [J]. arXiv Preprint, arXiv: 2105.14491, 2021.

[24] ZHAO S J, ZHANG X L, XIAO P F, et al. Exchanging dual-encoder–decoder: a new strategy for change detection with semantic guidance and spatial localization[J]. IEEE Transactions on Geoscience and Remote Sensing, 2023, 61: 1-16.

[25] DRAPER-GIL G, LASHKARI A H, MAMUN M S I, et al. Characterization of encrypted and VPN traffic using time-related features[C]// Proceedings of the 2nd International Conference on Information Systems Security and Privacy. Setúbal: SciTePress, 2016: 407-414.

[26] HABIBI LASHKARI A, DRAPER GIL G, MAMUN M S I, et al. Characterization of tor traffic using time based features[C]//Proceedings of the 3rd International Conference on Information Systems Security and Privacy. Setúbal: SciTePress, 2017: 253-262.

[27] EDE T V, BORTOLAMEOTTI R, CONTINELLA A, et al. FlowPrint: semi-supervised mobile-app fingerprinting on encrypted network traffic[C]//Proceedings 2020 Network and Distributed System Security Symposium. Piscataway: IEEE Press, 2020: 27.

[28] SIRINAM P, IMANI M, JUAREZ M, et al. Deep fingerprinting: undermining website fingerprinting defenses with deep learning[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2018: 1928-1943.

[29] WANG X, CHEN S H, SU J S. App-net: a hybrid neural network for encrypted mobile traffic classification[C]//Proceedings of the IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). Piscataway: IEEE Press, 2020: 424-429.

[30] ZHANG H, GOU G P, XIONG G, et al. Multi-granularity mobile encrypted traffic classification based on fusion features[C]//Science of Cyber Security: Third International Conference. Berlin: Springer, 2021: 154-170.

[31] ZHAO R J, ZHAN M W, DENG X W, et al. Yet another traffic classifier: a masked autoencoder based traffic transformer with multi-level flow representation[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2023: 5420-5427.

[32] HUOH T L, LUO Y, LI P L, et al. Flow-based encrypted network traffic classification with graph neural networks[J]. IEEE Transactions on Network and Service Management, 2023, 20(2): 1224-1237.

[33] ZHANG H Z, YUE H D, XIAO X, et al. Revolutionizing encrypted traffic classification with MH-net: a multi-view heterogeneous graph model [C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2025: 1048-1056.

[作者简介]

刘涛涛（1996–），男，江西吉安人，海军工程大学博士生，主要研究方向为人工智能、信息处理、网络安全。

付钰（1982–），女，湖北武汉人，博士，海军工程大学教授、博士生导师，主要研究方向为信息安全、人工智能。

俞艺涵（1992–），男，浙江金华人，博士，海军工程大学讲师，主要研究方向为隐私保护、信息安全。

安义帅（1997–），男，山西忻州人，海军工程大学博士生，主要研究方向为人工智能、网络安全。