Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

基于GNN的加密流量分类方法

硕士研究生 李国浩

2025年06月02日

问题回溯

- 总结反思
  - 声音太小
  - 论文算法原理不够深入

- 相关内容
  - 九尾狐
  - 2025.02.10 李国浩 《预训练加密流量分类方法》
  - 2023.08.07 巩锟 《预训练加密流量表征方法》

内容提要

- 预期收获
- 题目内涵解析
- 研究背景与意义
- 研究历史与现状
- 知识基础
- 算法原理
  - TFE-GNN
  - MH-Net
- 特点总结与知识展望
- 参考文献

预期收获

- 预期收获
  - 1.了解加密流量分类的基本概念和研究方向
  - 2.理解两种基于GNN的流量分类方法的基本原理
  - 3.了解现有方法的缺陷及未来的发展方向

题目内涵解析

- 内涵解析
  - 加密流量：由加密算法生成的流量，主要是指在通信过程中所传送的被加密过的实际明文内容
  - 识别结果：加密与非加密，协议，应用，服务等
- 研究目标
  - 利用深度学习等方法，分析网络流量（即使其内容被加密）来识别流量的类型、应用或行为，绕过对于加密流量解密，从而支持网络管理、安全防护和资源优化

1	0.000000000	127.0.0.1	127.0.0.1	TCP	74	34012 → 4443 [SYN
2	0.000032000	127.0.0.1	127.0.0.1	TCP	74	4443 → 34012 [SYN
3	0.000052000	127.0.0.1	127.0.0.1	TCP	66	34012 → 4443 [ACK
4	0.000304083	127.0.0.1	127.0.0.1	TLSv1.3	281	Client Hello
5	0.000310708	127.0.0.1	127.0.0.1	TCP	66	4443 → 34012 [ACK
6	0.007317708	127.0.0.1	127.0.0.1	TLSv1.3	1403	Server Hello, Cha
7	0.007344791	127.0.0.1	127.0.0.1	TCP	66	34012 → 4443 [ACK
8	0.008424750	127.0.0.1	127.0.0.1	TLSv1.3	146	Change Cipher Spe
9	0.008440541	127.0.0.1	127.0.0.1	TCP	66	4443 → 34012 [ACK
10	0.008914625	127.0.0.1	127.0.0.1	TLSv1.3	103	Application Data

题目内涵解析

- 研究背景
  - 随着互联网的快速发展，HTTPS/TLS 成为主流，VPN、Tor 和代理的流行，新兴加密协议的出现，网络流量加密已成为保护用户隐私和数据安全的重要手段
  - 传统流量分类方法的失效：基于端口的方法失效，深度包检测方法受限，机器学习方法依赖特征工程
- 研究意义
  - 网络管理与优化（优化带宽分配，限制非业务流量）
  - 安全防护与威胁检测（恶意软件检测，入侵检测）
  - 隐私保护与合规监管的平衡（在不解密的情况下分析流量，识别非法活动）

研究历史与现状

2017 Wang等人首次使用一个基于CNN的端到端框架对恶意软件加密流量分析，识别加密流量属于那一恶意软件种类，将表征学习引入加密流量分类(ETC)领域

2018 Lotfollahi等人设计一个嵌入堆叠式自动编码器和卷积神经网络的模型，用于对网络流量进行分类，标志深度学习方法在ETC领域的成熟化应用

2021 Shen等人提出了一种名为流量交互图的图结构作为加密应用流的信息表示，将 DApp 指纹识别转化为图分类问题，设计了一个基于GNN的分类器对流量交互图分类

2022 Lin等人设计加密流量表征模型(ET-BERT),将加密流量转换为BURST结构,设计两种预训练方法,通过微调预训练好的模型即可实现不同场景下的流量类型识别

2023 Zhao等人提出了一种多级流表示结构多级流表示，利用一个掩蔽自动编码器（MAE）对多级流表示分类，在低复杂度的情况下达到较好的效果

2023 Zhang等人提出了一种基于逐点互信息（PMI）的字节级流量图构建方法，以及一种使用图神经网络（TFE-GNN）进行特征提取的时间融合编码器模型

2025 Zhang等人将不同数量的流量比特聚合为多种类型的流量单元，从而构建具有不同信息粒度的多视图流量图,采用对比学习来加强所学习的流量单元表示的稳健性

- 流量粒度
  - 包级别（Packet）
    - 头部（header）：包含MAC，IP，传输层控制信息等内容
    - 有效负载（payload）：传输层以上的数据总和
  - 流级别（Flow，单向流）
    - 五元组：(源IP, 源端口,目的IP, 目的端口, 传输层协议)
  - 会话级别（Session，双向流）
    - （源IP, 源端口）和（目的IP, 目的端口）互换，传输层协议不变
  - 主机级别（Host）
  - 分割工具：SplitCap

```
> Frame 19: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface lo, id 0
> Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 34012, Dst Port: 4443, Seq: 333, Ack: 6118, Len: 24
> Transport Layer Security
```

知识基础

- 点互信息（PMI）
  - 衡量两个事件之间关联强度的指标，评估两个具体事件（如单词、符号等）的共现概率与它们独立出现概率的关系

$$PMI(x,y) = \frac{P(x,y)}{P(x) \cdot P(y)}$$

- 图神经网络（GNN）
  - 通过消息传递（Message Passing）机制，让节点之间交换信息并更新自身表示
    - 聚合（Aggregate）：每个节点收集邻居节点的信息
    - 更新（Update）：结合自身和邻居信息，生成新的节点表示
    - 循环迭代：多次重复上述步骤，使信息在整个图中传播

$$h_v^{(k)} = update(h_v^{(k-1)}, aggregate(\{h_u^{(k-1)} \mid u \in \mathcal{N}(v)\}))$$

算法一 TFE-GNN

TFE-GNN: A Temporal Fusion Encoder Using Graph Neural Networks for Fine-grained Encrypted Traffic Classification

目标与难点

T	目标	利用加密流量中的头部和负载信息实现服务类型和应用分类
I	输入	ISCX VPN-nonVPN数据集，ISCX Tor-nonTor数据集，WWT数据集
P	处理	1.将数据集以session形式，划分为多条双向流 2.将头部和负载分开，根据PMI计算包中字节相关性，将数据包转为流量图表示 3.将流量图输入流量图编码器中，得到每个数据包的整体表征 4.将包的整体表征应用于下游分类任务
O	输出	服务类型分类与应用分类结果

P	问题	1.现有方法平等地对待数据包的头部和有效负载，但忽略了它们之间含义的差异 2.现有方法将其原始字节作为节点特征，没有充分利用原始字节
C	条件	预先对数据集做出流分割
D	难点	如何根据包中字节构建流量图表示
L	水平	2023 CCF-A (WWW会议)

算法原理

- 数据集预处理
  - 将原始数据集首先按照服务类型或应用类型划分到不同的文件夹中，以session形式将大pcap文件划分小pcap文件

```
SplitCap.exe -r large.pcap -s session -o "D:\sessions\"
```

- 字节级流量图构建
  - 核心思想：一个包对应于一个图，包中的字节对应于图中的节点，节点数目不超过256个，节点特征为原始字节值
  - 字节之间的相关性表示：采用逐点互信息（PMI）来对两个字节之间的相关性进行建模
  - 边缘创建：正的PMI值意味着字节的语义相关性很高，而零或负的PMI值意味着字节的语义相关性很小或没有，只在PMI值为正的两个字节之间创建一条边

算法原理

- 字节级流量图构建
  - 图构建
    - 通过节点 i 和 j 的邻接矩阵 A 的给出边的创建形式描述

$$a_{ij} = \begin{cases} 1, & \text{PMI}(i,j) > 0 \\ 0, & \text{Otherwise} \end{cases}$$

    - 图 $\mathcal{G}$ 中每个节点的初始特征由相应的字节值给出，范围从0到255（i.e. 00-ff）
    - 由于$PMI(i,j) = PMI(j,i)$，因此字节级流量图是无向的
    - 上述过程分别对header和payload构建得到$\mathcal{G}_h, \mathcal{G}_p$

```
00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00
10 f1 ba f8 40 00 40 06  71 0c 7f 00 00 01 7f 00
00 01 11 5b 84 dc eb bc  bb 52 d5 b5 05 47 80 18
02 00 0e e6 00 00 01 01  08 0a 64 b0 97 d0 64 b0
97 d0 17 03 03 10 b8 d3  8b 34 e7 96 51 b9 ac 6e
93 94 b9 6a c3 d2 4f 2d  74 09 d8 bc 51 fd b7 78
3b d6 96 08 99 77 f1 41  be 20 a7 b3 c2 d0 a7 44
ab fd 4a a1 b1 54 02 08  88 d9 d2 43 c3 6d f4 59
da 7f 91 3b 72 b8 71 96  3d f1 66 5b 75 e2 12 84
```

算法原理

- 双嵌入
  - 采用双嵌入原因：数据包的header和payload中分别具有相同值的两个字节的表示含义可能完全不同，payload携带数据包的传输内容，header是携带的是数据包的控制内容
  - 单嵌入结果：如果payload和header中具有相同值的两个字节对应于相同的嵌入，那么由于含义模糊，模型很难在这些嵌入参数上收敛到最佳值
  - 采用具有两个不共享参数的嵌入层，将图中节点的初始字节值特征分别嵌入到两种多维嵌入向量中,嵌入矩阵如下：

$$\begin{cases} E_{header} \in \mathbb{R}^{K\times d} \\ E_{payload} \in \mathbb{R}^{K\times d} \end{cases}$$

    - 其中K为嵌入元素的数目，d为嵌入维度

算法原理

- 带有交叉门控特征融合的流量图编码器
  - 使用堆叠的GraphSAGE作为流量图编码器的主干，处理流程如下：
    - 首先，GraphSAGE通过使用节点v的度对每个邻近节点u的嵌入进行归一化，来计算来自每个邻近节点$u \in N(v)$的消息
    - 其次，通过逐元素平均值操作计算所有邻近节点$N(v)$的总体消息，并通过拼接操作，聚合总体消息以及节点v的嵌入，得到新嵌入
    - 然后，对节点v的嵌入进行非线性变换，完成一个GraphSAGE层的正向过程

$$\mathbf{m}_{N(v)}^{(l)} = \sum_{u \in N(v)} \frac{\mathbf{h}_u^{(l-1)}}{|N(v)|}$$

$$\mathbf{h}_v^{(l)} = \sigma\left(\mathbf{w}^{(l)} \cdot \mathrm{CONCAT}\left(\mathbf{h}_v^{(l-1)}, \mathbf{m}_{N(v)}^{(l)}\right)\right)$$

算法原理

- 带有交叉门控特征融合的流量图编码器
  - 使用堆叠的GraphSAGE作为流量图编码器的主干

$$\mathbf{m}_{N(v)}^{(l)} = \sum_{u \in N(v)} \frac{\mathbf{h}_u^{(l-1)}}{|N(v)|}$$

$$\mathbf{h}_v^{(l)} = \sigma\left(\mathbf{w}^{(l)} \cdot \mathrm{CONCAT}\left(\mathbf{h}_v^{(l-1)}, \mathbf{m}_{N(v)}^{(l)}\right)\right)$$

- $w(l) \in \mathbb{R}^{d_{l-1} \times d_l}$是第$l$层中的参数矩阵，CONCAT（·）表拼接运算，σ（·）表示激活函数。特别地，使用参数化ReLU（PReLU）作为激活函数。
- 采用PRELU的原因：PReLU通过一个因子对每个负元素值进行缩放，这不仅起到非线性变换的作用，而且通过负轴上的每个通道的不同缩放因子发挥类似于注意力机制的作用
- 最后，通过批量归一化（BN）对更新后的特征$h_v^{(l)}$进行归一化

算法原理

- 带有交叉门控特征融合的流量图编码器
  - 使用堆叠的GraphSAGE作为流量图编码器的主干
    - 深度GNN模型中的过度平滑问题，只堆叠GraphSAGE最多4层，并级联输出每个节点v的每个层的输出特征，这种形式类似于跳跃知识网络（JKN）

$$\mathbf{h}_v^{\text{final}} = \text{CONCAT}(\mathbf{h}_v^{(1)}, \mathbf{h}_v^{(2)}, \mathbf{h}_v^{(3)}, \mathbf{h}_v^{(4)})$$

      - 其中h_v^{final}是节点v的最终特征向量
    - 最后，在所有节点的最终输出向量上应用均值池化以获得图特征向量g

$$\mathbf{g} = \frac{\mathbf{h}_1^{\text{final}} \oplus \ldots \oplus \mathbf{h}_{|\mathcal{V}|}^{\text{final}}}{|\mathcal{V}|}$$

      - 其中，⊕表示逐元素加法
    - g_h和g_p来表示分别从流量header图和流量payload图中提取的特征向量

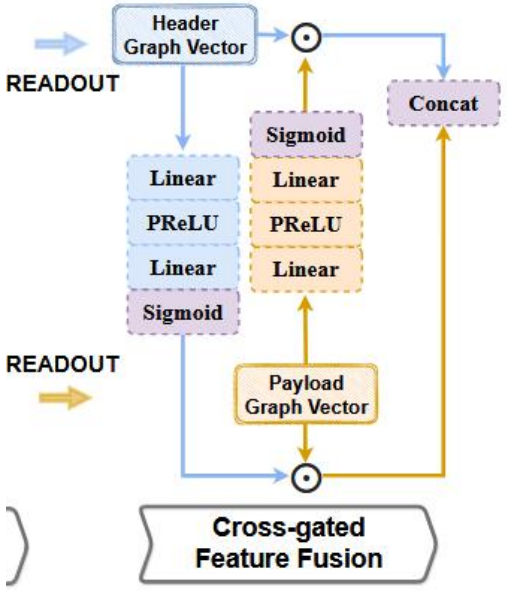
算法原理

- 带有交叉门控特征融合的流量图编码器
  - 交叉门控特征融合
    - 目标：在g_h和g_p之间创建合理的关系，以获得数据包字节的整体表示
    - 结构：采用两个过滤器，每个过滤器由两个线性层组成，它们之间有一个PReLU激活函数
      - 将不共享参数的两个过滤器分别应用于g_h和g_p
      - 然后使用逐元素Sigmoid函数将每个元素缩放为[0,1]
      - 将缩放后的向量视为门控向量（头部和有效负载的s_h和s_p），并使用它们来对相应的g_h和g_p进行量化过滤

$$s_h = \text{Sigmoid}(w_{h2}^T \text{PReLU}(w_{h1}^T g_h + b_{h1}) + b_{h2})$$

$$s_p = \text{Sigmoid}(w_{p2}^T \text{PReLU}(w_{p1}^T g_p + b_{p1}) + b_{p2})$$

$$z = \text{CONCAT}(s_h \odot g_p, s_p \odot g_h)$$



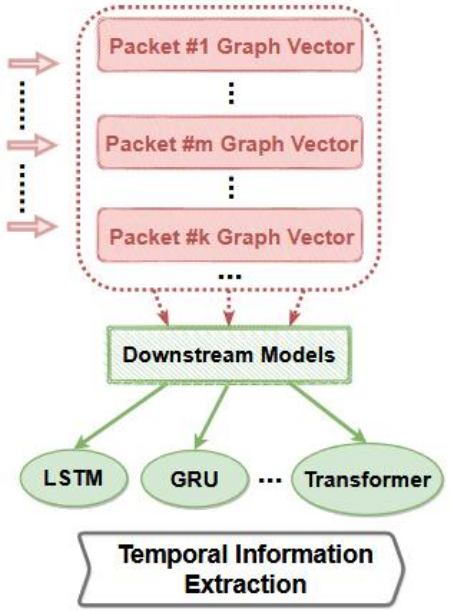
  - 其中，符号⊙表示逐元素积，z是数据包的总体表示向量，可用于下游任务

算法原理

- 下游任务端到端训练
  - 时间信息提取
    - 由于将流中每个数据包的原始字节编码到表示向量z中，因此流级分类任务可以被视为时间序列预测任务
    - 采用长短期记忆（LSTM）作为下游基线模型，LSTM是双向的，有两层，其输出向量被馈送到一个两层线性分类器中，PReLU作为其激活函数，以获得最终的预测结果

$$\mathcal{L}_{TFE-GNN} = \mathrm{CE}(\mathrm{Classifier}(\mathrm{LSTM}(\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n)), y)$$

    - 其中，n是流长度，y是真实标签，CE（·）表示交叉熵函数



实验设计

- 实验设置
  - 数据集： 公开的ISCX VPN-nonVPN、ISCX Tor-nonTor和自行收集的WWT数据集
  - 预处理：
    - 定义并过滤出两种“异常”样本：（1）空流或空段：所有数据包都没有有效载荷的流量流或段；（2）超长流或段：长度（即数据包数量）大于10000的流量流或段
    - 删除MAC，源和目标IP地址以及端口号
  - 实施细节：
    - 将一个样本的最大数据包数量设置为50，最大payload字节长度和最大header字节长度分别设置为150和40，PMI窗口大小设置为5
  - 指标：总体准确度（AC）、精确度（PR）、召回率（RC）和宏观F1分数（F1）
  - 基线：传统的基于特征工程的方法、基于深度学习的方法、以及基于图神经网络的方法

实验设计

- 对比实验（WWT数据集）
  - 在WWT数据集上与的几个基线相比，TFE-GNN达到了最佳性能
  - 几乎所有的基线在Telegram数据集上的表现都很差，这是由于VPN的使用增加了分类难度，并在VPN造成的暂时不良网络条件下引入了一些背景噪声

Dataset	WeChat				WhatsApp				Telegram			
Model	AC	PR	RC	F1	AC	PR	RC	F1	AC	PR	RC	F1
AppScanner [31]	0.9927	0.9908	0.9904	0.9905	0.9790	0.9688	0.9601	0.9628	0.8379	0.8154	0.8653	0.8304
K-FP [8]	0.9741	0.9665	0.9630	0.9645	0.9710	0.9589	0.9515	0.9526	0.8797	0.8378	0.8990	0.8567
FlowPrint [32]	0.7429	0.5302	0.6380	0.5532	0.6197	0.3820	0.4934	0.3880	0.4833	0.4025	0.5000	0.4311
CUMUL [23]	0.9472	0.9497	0.9412	0.9390	0.9855	0.9835	0.9699	0.9756	0.8330	0.7980	0.8471	0.8053
GRAIN [43]	0.9404	0.9366	0.9369	0.9251	0.9724	0.9685	0.9475	0.9550	0.8003	0.7615	0.7903	0.7407
FAAR [19]	0.9873	0.9863	0.9847	0.9854	0.9790	0.9683	0.9566	0.9597	0.8184	0.7884	0.8507	0.8018
ETC-PS [40]	0.9863	0.9864	0.9831	0.9846	0.9833	0.9743	0.9685	0.9703	0.8477	0.8295	0.8710	0.8382
FS-Net [18]	0.9223	0.9446	0.9196	0.8964	0.9942	0.9949	0.9919	0.9933	0.8469	0.8161	0.8744	0.8272
DF [30]	0.9800	0.9757	0.9751	0.9751	0.8978	0.7954	0.8406	0.8143	0.8251	0.8206	0.8542	0.7960
EDC [16]	0.9746	0.9653	0.9625	0.9620	0.9732	0.9589	0.9526	0.9546	0.8153	0.8118	0.8372	0.7896
FFB [44]	0.9675	0.9691	0.9661	0.9644	0.9350	0.8957	0.8767	0.8708	0.7990	0.7630	0.8169	0.7802
MVML [4]	0.9643	0.9519	0.9540	0.9526	0.9456	0.9302	0.9009	0.8971	0.7943	0.7413	0.7636	0.7340
ET-BERT [17]	0.9505	0.9368	0.9285	0.9266	0.9107	0.8934	0.8697	0.8439	0.7679	0.8712	0.7989	0.7858
GraphDApp [29]	0.8763	0.8368	0.8378	0.8330	0.8753	0.7615	0.8060	0.7791	0.7766	0.7575	0.7627	0.7227
ECD-GNN [11]	0.9863	0.9847	0.9830	0.9835	0.9811	0.9722	0.9647	0.9672	0.1810	0.0353	0.1641	0.0528
TFE-GNN	**0.9956**	**0.9953**	**0.9939**	**0.9946**	**0.9971**	**0.9957**	**0.9966**	**0.9961**	**0.9586**	**0.9584**	**0.9742**	**0.9649**

实验设计

- 对比实验（ ISCX VPN-nonVPN， ISCX Tor-nonTor ）
  - TFE-GNN优于公共数据集上的几乎所有基线方法，但在非VPN数据集，TFE-NNN和ET-BERT都达到了类似的结果
  - 然而，ET-BERT是一个具有非常复杂模型架构的大型模型，而TFE-GNN是一个较小的模型

Dataset	ISCX-VPN				ISCX-nonVPN				ISCX-Tor				ISCX-nonTor			
Model	AC	PR	RC	F1	AC	PR	RC	F1	AC	PR	RC	F1	AC	PR	RC	F1
AppScanner [31]	0.8889	0.8679	0.8815	0.8722	0.7576	0.7594	7465	0.7486	0.7543	0.6629	0.6042	0.6163	0.9153	0.8435	0.8140	0.8273
K-FP [8]	0.8713	0.8750	0.8748	0.8747	0.7551	0.7478	0.7354	0.7387	0.7771	0.7417	0.6209	0.6313	0.8741	0.8653	0.7792	0.8167
FlowPrint [32]	0.8538	0.7451	0.7917	0.7566	0.6944	0.7073	0.7310	0.7131	0.2400	0.0300	0.1250	0.0484	0.5243	0.7590	0.6074	0.6153
CUMUL [23]	0.7661	0.7531	0.7852	0.7644	0.6187	0.5941	0.5971	0.5897	0.6686	0.5349	0.4899	0.4997	0.8605	0.8143	0.7393	0.7627
GRAIN [43]	0.8129	0.8077	0.8109	0.8027	0.6667	0.6532	0.6664	0.6567	0.6914	0.5253	0.5346	0.5234	0.7895	0.6714	0.6615	0.6613
FAAR [19]	0.8363	0.8224	0.8404	0.8291	0.7374	0.7509	0.7121	0.7252	0.6971	0.5915	0.4876	0.4814	0.9103	0.8253	0.7755	0.7959
ETC-PS [40]	0.8889	0.8803	0.8937	0.8851	0.7273	0.7414	0.7133	0.7208	0.7486	0.6811	0.5929	0.6033	0.9365	0.8700	0.8311	0.8486
FS-Net [18]	0.9298	0.9263	0.9211	0.9234	0.7626	0.7685	0.7534	0.7555	0.8286	0.7487	0.7197	0.7242	0.9278	0.8368	0.8254	0.8285
DF [30]	0.8012	0.7799	0.8152	0.7921	0.6742	0.6857	0.6717	0.6701	0.6514	0.4803	0.4767	0.4719	0.8568	0.8003	0.7415	0.7590
EDC [16]	0.7836	0.7747	0.8108	0.7888	0.6970	0.7153	0.7000	0.6978	0.6400	0.4980	0.4528	0.4504	0.8692	0.7994	0.7411	0.7451
FFB [44]	0.8304	0.8714	0.8149	0.8335	0.7020	0.7274	0.6945	0.7050	0.6343	0.4870	0.5203	0.4952	0.8954	0.7545	0.7430	0.7430
MVML [4]	0.6491	0.7231	0.6198	0.6151	0.5126	0.5751	0.4707	0.4806	0.6343	0.3914	0.4104	0.3752	0.7235	0.5488	0.5512	0.5457
ET-BERT [17]	0.9532	0.9436	0.9507	0.9463	**0.9167**	0.9245	**0.9229**	0.9235	0.9543	0.9242	0.9606	0.9397	0.9029	0.8560	0.8217	0.8332
GraphDApp [29]	0.6491	0.5668	0.6103	0.5740	0.4495	0.4230	0.3647	0.3614	0.4286	0.2557	0.2509	0.2281	0.6936	0.5447	0.5398	0.5352
ECD-GNN [11]	0.1111	0.0185	0.1667	0.0333	0.0606	0.0101	0.1667	0.0190	0.0571	0.0071	0.1250	0.0135	0.9078	0.8015	0.8168	0.7977
TFE-GNN	**0.9591**	**0.9526**	**0.9593**	**0.9536**	0.9040	**0.9316**	0.9190	**0.9240**	**0.9886**	**0.9792**	**0.9939**	**0.9855**	**0.9390**	**0.8742**	**0.8335**	**0.8507**

实验设计

- 消融实验（ISCX-VPN，ISCX-Tor）
  - 模块标记：将头部、负载、双嵌入模块、跳跃式知识网络连接、交叉门控特征融合和激活函数与批归一化分别表示为H, P, dual, JKN, CGFF, A&N
  - 结论：
    - 数据包报头在分类中的作用比数据包有效载荷更重要，不同的数据集具有不同级别的报头和有效载荷重要性
    - 双嵌入的使用使F1得分分别提高了3.63%和0.95%，表明其总体有效性。类似JKN的级联和交叉门控特征融合在两个数据集上都以相似的幅度提高了TFE-GNN的性能

Method	H	P	DUAL	JKN	CGFF	A&N	AC	PR	RC	F1
w/o P	✓	×	×	✓	×	✓	0.8713 | 0.9874	0.8261 | 0.9788	0.8228 | 0.9934	0.8230 | 0.9806
w/o H	×	✓	×	✓	×	✓	0.8051 | 0.8726	0.8232 | 0.7780	0.7957 | 0.7809	0.7980 | 0.7700
w/o DUAL	✓	✓	×	✓	✓	✓	0.9310 | 0.9829	0.9149 | 0.9747	0.9209 | 0.9801	0.9173 | 0.9760
w/o JKN	✓	✓	✓	×	✓	✓	0.9474 | 0.9790	0.9365 | 0.9736	0.9397 | 0.9879	0.9374 | 0.9795
w/o CGFF	✓	✓	✓	✓	×	✓	0.9445 | 0.9800	0.9329 | 0.9717	0.9371 | 0.9847	0.9339 | 0.9770
w/o A&N	✓	✓	✓	✓	✓	×	0.6105 | 0.2212	0.5576 | 0.0555	0.5487 | 0.1180	0.5289 | 0.0548
w/ SUM	✓	✓	✓	✓	✓	✓	0.8497 | 0.8194	0.8549 | 0.7287	0.8380 | 0.6986	0.8426 | 0.6891
w/ MAX	✓	✓	✓	✓	✓	✓	0.8480 | 0.9870	0.8328 | 0.9752	0.8115 | 0.9778	0.8094 | 0.9751
w/ GRU	✓	✓	✓	✓	✓	✓	0.8550 | 0.8932	0.8489 | 0.8702	0.8287 | 0.8664	0.8294 | 0.8610
w/ TRANSFORMER	✓	✓	✓	✓	✓	✓	0.6754 | 0.9777	0.5706 | 0.9753	0.5992 | 0.9820	0.5658 | 0.9828
TFE-GNN (default)	✓	✓	✓	✓	✓	✓	**0.9591 | 0.9886**	**0.9526 | 0.9792**	**0.9593 | 0.9939**	**0.9536 | 0.9855**

实验设计

- 消融实验(ISCX-VPN，ISCX-Tor)
  - 结论：
    - 在两个数据集上，节点特征的SUM运算比MEAN运算在F1上差11.1%和29.64%，在ISCX-VPN数据集上，节点特征的MAX运算F1得分降低到更差的结果，而在ISCX-Tor数据集上仅略微降低F1得分
    - 将默认的下游模型LSTM更改为GRU，由于GRU的架构更简单，这会使两个数据集上的所有指标恶化约10%；此外，采用transformer作为下游模型,结果表明，transformer在ISCX-Tor上表现良好，而在ISCX-VPN上F1下降了近40%

Method	H	P	DUAL	JKN	CGFF	A&N	AC	PR	RC	F1
w/o P	✓	×	×	✓	×	✓	0.8713 | 0.9874	0.8261 | 0.9788	0.8228 | 0.9934	0.8230 | 0.9806
w/o H	×	✓	×	✓	×	✓	0.8051 | 0.8726	0.8232 | 0.7780	0.7957 | 0.7809	0.7980 | 0.7700
w/o DUAL	✓	✓	×	✓	✓	✓	0.9310 | 0.9829	0.9149 | 0.9747	0.9209 | 0.9801	0.9173 | 0.9760
w/o JKN	✓	✓	✓	×	✓	✓	0.9474 | 0.9790	0.9365 | 0.9736	0.9397 | 0.9879	0.9374 | 0.9795
w/o CGFF	✓	✓	✓	✓	×	✓	0.9445 | 0.9800	0.9329 | 0.9717	0.9371 | 0.9847	0.9339 | 0.9770
w/o A&N	✓	✓	✓	✓	✓	×	0.6105 | 0.2212	0.5576 | 0.0555	0.5487 | 0.1180	0.5289 | 0.0548
w/ SUM	✓	✓	✓	✓	✓	✓	0.8497 | 0.8194	0.8549 | 0.7287	0.8380 | 0.6986	0.8426 | 0.6891
w/ MAX	✓	✓	✓	✓	✓	✓	0.8480 | 0.9870	0.8328 | 0.9752	0.8115 | 0.9778	0.8094 | 0.9751
w/ GRU	✓	✓	✓	✓	✓	✓	0.8550 | 0.8932	0.8489 | 0.8702	0.8287 | 0.8664	0.8294 | 0.8610
w/ TRANSFORMER	✓	✓	✓	✓	✓	✓	0.6754 | 0.9777	0.5706 | 0.9753	0.5992 | 0.9820	0.5658 | 0.9828
TFE-GNN (default)	✓	✓	✓	✓	✓	✓	**0.9591 | 0.9886**	**0.9526 | 0.9792**	**0.9593 | 0.9939**	**0.9536 | 0.9855**

算法原理

- 算法总结
  - 算法贡献
    - 提出了一种字节级流量图构建的方法和一个名为TFE-GNN的加密流量分类模型
      - 字节级流量图构建方法可以挖掘原始字节之间的潜在相关性，并生成有区别的流量图
      - TFE-GNN旨在从构建的流量图中提取高维特征
  - 算法不足
    - 有限图构造方法：在训练过程之前确定所提出模型的图拓扑，这可能会导致非最优性能
    - 字节序列中隐含未使用的时间信息：字节级流量图是在不引入字节序列的显式时间特征的情况下构建的

算法原理 MH-Net

Revolutionizing Encrypted Traffic Classification with MH-Net: A Multi-View Heterogeneous Graph Model

目标与难点

T	目标	利用不同流量单元和不同字节相关性进行加密流量服务类型分类
I	输入	ISCX VPN-nonVPN数据集，ISCX Tor-nonTor数据集， CIC-IoT数据集
P	处理	1.将不同数量的流量比特聚合到多种类型的流量单元中，预备构建多视图流量图 2.利用流量单元三种相关性来模拟流量图的异质性，构建多视角异构流量图 3.采用异质图神经网络进行特征提取 4.以多任务的方式进行对比学习
O	输出	加密流量不同服务类型分类结果

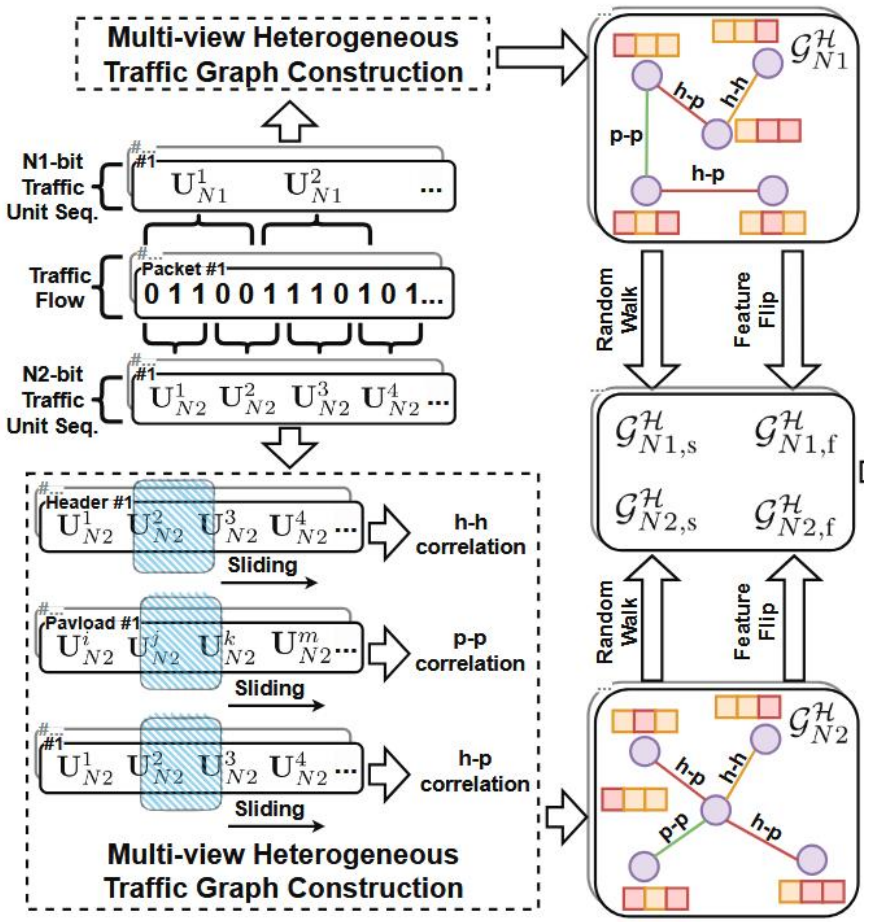
P	问题	1.现有方法将字节视为不可分割的单元，忽略了流量数据中包含不同粒度的信息 2.现有方法混合了字节序列中不同位置的字节相关性，忽略了不同类型相关性之间的差异
C	条件	预先对数据集做出流分割
D	难点	多视角异构流量图的构建
L	水平	2025 CCF-A (AAAI会议)

算法原理

- 多视图流量图构建
  - 具有不同粒度的流量单元概念
    - 将不同长度的原始比特聚合到各种流量单元中（U_N），具有不同比特数的流量单元可以从不同的角度“解释”或“表达”传输的数据，从而能够挖掘潜在特征
  - 流量图构建：将不同长度的流量单元序列转化为流量图，利用点互信息（PMI）来量化流量单元之间的相关性
    - 如果$PMI(U_N^i, U_N^j) > 0$，表明流量单元相关性高
    - 对于每个数据包中的流量数据，选择两种类型的流量单元（U_{N_1}, U_{N_2}），并在它们的序列上应用PMI算法，分别构建每个数据包的多视图流量图$(\mathcal{G}_{N_1}, \mathcal{G}_{N_1})$
    - 流量图的节点特征是每个流量单元的值，当$N < 8$时，分割后的单元做补零处理

算法原理

- 异构流量单元相关性分析
  - 功能不同：报头和有效载荷由于功能不同而具有信息异构性（前者携带数据包的元数据，后者携带实际传输的内容）
  - 三种相关性：提出三种类型的流量单元相关性，即报头–报头（h-h）、有效载荷–有效载荷（p-p）和报头–有效载荷（h-p）相关性
  - 将PMI算法分别应用于h-h 、 p-p和h-p的流量单元序列，以获得三种类型的流量单元连接，将这些连接进一步集成为异构流量图 $\mathcal{G}_N^{\mathcal{H}}$ ，从而将多视角流量图 $(\mathcal{G}_{N_1}, \mathcal{G}_{N_1})$转换为多视图异构流量图$(\mathcal{G}_{N_1}^{\mathcal{H}}, \mathcal{G}_{N_2}^{\mathcal{H}})$



算法原理

- 异构流量图编码器
  - 目标：将每个数据包的流量图($\mathcal{G}_{N_1}^{\mathcal{H}}$, $\mathcal{G}_{N_2}^{\mathcal{H}}$) 馈送到异构流量图编码器中，用于流量表示学习
  - 结构：异构流量图编码器采用异构图神经网络（HGNN）来提取流量图的判别特征，采用GraphSAGE作为其基本骨干

$$\mathbf{m}_v^{(l)} = \mathrm{MSG}^{(l)}\left(\left\{\mathbf{h}_u^{(l-1)}, u \in N(v)\right\}; \theta_{h\text{-}h}^{l,m}, \theta_{p\text{-}p}^{l,m}, \theta_{h\text{-}p}^{l,m}\right) \tag{3}$$

$$\mathbf{h}_v^{(l)} = \mathrm{AGG}^{(l)}\left(\mathbf{h}_v^{(l-1)}, \mathbf{m}_v^{(l)}; \theta_{h\text{-}h}^{l,a}, \theta_{p\text{-}p}^{l,a}, \theta_{h\text{-}p}^{l,a}\right) \tag{4}$$

  - 对最后一层GraphSAGE的所有节点嵌入向量执行元素平均运算，以分别获得流量图($\mathcal{G}_{N_1}^{\mathcal{H}}$, $\mathcal{G}_{N_2}^{\mathcal{H}}$)的最终包级流量表示(p_{N_1}, p_{N_2})

$$\mathbf{p}_{N1} = \mathrm{HGNN}(\mathcal{G}_{N1}^{\mathcal{H}}), \quad \mathbf{p}_{N2} = \mathrm{HGNN}(\mathcal{G}_{N2}^{\mathcal{H}})$$

  - 可以使用循环神经网络（RNN）进一步获得流级流量表示(f_{N_1}, f_{N_2})

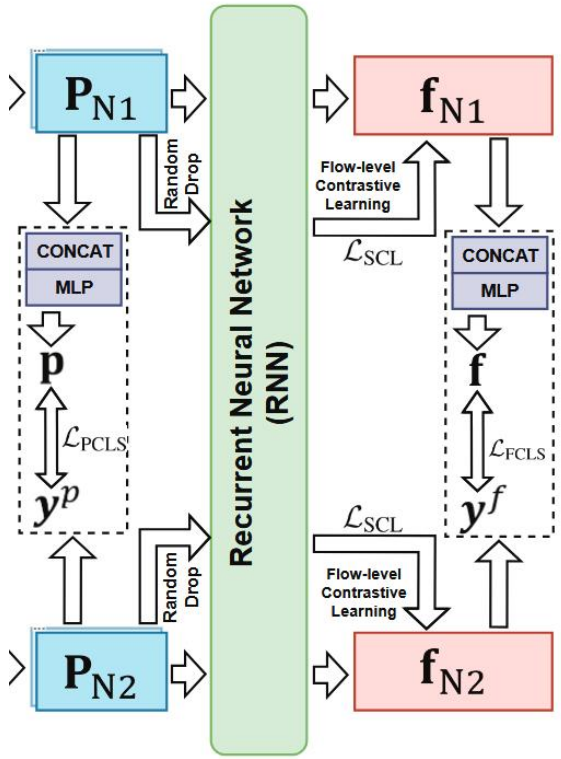
算法原理

- 多任务训练
  - 训练任务主要包括流量分类任务和对比学习任务
  - 流量分类任务
    - 同时在MH-Net中执行流级和包级流量分类任务
    - 使用一个非共享参数的多层感知器（MLP）来转换流级和数据包级表示，并分别计算相应的流量分类任务损失

$$\mathbf{f} = \mathrm{CONCAT}(\mathbf{f}_{N1}, \mathbf{f}_{N2}), \quad \mathbf{p} = \mathrm{CONCAT}(\mathbf{p}_{N1}, \mathbf{p}_{N2}) \tag{7}$$

$$\mathcal{L}_{\mathrm{FCLS}} = \mathrm{CE}(\mathrm{MLP}(\mathbf{f}), \boldsymbol{y}^f), \quad \mathcal{L}_{\mathrm{PCLS}} = \mathrm{CE}(\mathrm{MLP}(\mathbf{p}), \boldsymbol{y}^p) \tag{8}$$

    - 其中CONCAT（·）表示拼接操作，CE（·）是交叉熵损失函数，y^f是流标签，y^p是与其所属流标签一致的数据包标签



算法原理

- 多任务训练
  - 双层次对比学习任务：包级对比学习，流级对比学习
    - 包级对比学习
      - 由于包级流量表示是基于流量图得出的，需要生成一个增强图以进行进一步的对比学习
      - 主要采用了两种增强方法：图结构和节点特征增强
        - » **图结构增强**：利用随机游走算法，获得增强的流量图 $\mathcal{G}_{N,s}^{\mathcal{H}}$
        - » **节点特征增强**：通过翻转其节点特征来增强流量图，得到另一个增强的流量图 $\mathcal{G}_{N,f}^{\mathcal{H}}$
      - 在获得 $\mathcal{G}_{N,s}^{\mathcal{H}}$ 和 $\mathcal{G}_{N,f}^{\mathcal{H}}$ 的嵌入向量后，包级对比损失可以公式化为

$$\mathcal{L}_{\mathrm{PCL}} = \sum_{i}^{\{1,2\}} \mathcal{L}_{\mathrm{SCL}}(\{\mathrm{HGNN}(\mathcal{G}_{Ni,\mathrm{s}}^{\mathcal{H}}), \mathbf{p}_{Ni}\}) + \mathcal{L}_{\mathrm{SCL}}(\{\mathrm{HGNN}(\mathcal{G}_{Ni,\mathrm{f}}^{\mathcal{H}}), \mathbf{p}_{Ni}\})$$

算法原理

- 多任务训练
  - 双层次对比学习
    - 流级对比学习
      - 从流量流中按照一定概率和比例随机丢弃数据包来获得增强的流量流
      - 流级别对比损失可以公式化为：

$$\mathcal{L}_{\mathrm{FCL}} = \sum_{i}^{\{1,2\}} \mathcal{L}_{\mathrm{SCL}}(\{\mathrm{RNN}(\mathbf{p}_{Ni}^{1} \odot \rho_1, ..., \mathbf{p}_{Ni}^{L} \odot \rho_L), \mathbf{f}_{Ni}\})$$

» 其中，$\rho_i \in \{0,1\}$，服从伯努利分布，表示是否丢弃数据包

- 总体训练目标 $\mathcal{L} = \mathcal{L}_{\mathrm{PCLS}} + \mathcal{L}_{\mathrm{FCLS}} + \alpha\mathcal{L}_{\mathrm{PCL}} + \beta\mathcal{L}_{\mathrm{FCL}}$

  - 其中$\alpha, \beta \in [0,1]$，分别是控制包级和流级对比任务的贡献系数

实验设计

- 实验设置
  - 数据集：公开CIC-IoT数据集、ISCX VPN-nonVPN数据集和ISCX Tor-nonTor数据集
  - 实施细节
    - 利用4位和8位流量单元构建多视图异构流量图，实现多样性和计算成本之间的权衡
    - 在HGNN中， GraphSAGE层数设置为4；包转换为流的RNN初始化为LSTM；在对比学习中，丢包率比例设置为0.6，温度系数τ为0.07；总体损失中，目标系数 α 和 β 分别设置为1.0和0.5
  - 评估指标：总体准确度（AC）和宏观F1分数（F1）
  - 基线：流级别流量分类方法，包级别流量分类方法

实验设计

- 对比实验（流级对比实验）
  - MH-Net在CIC-IoT和ISCX数据集的两个指标上取得了总体最佳结果，其次是TFE-GNN和YaTC
  - 在这些方法中，尽管TFE-GNN和YaTC利用原始字节进行学习，但总体性能明显低于MH-Net，这意味着它们的模型设计中字节利用率不足，忽视了细粒度字节相关性
  - ET-BERT具有大量参数和大规模数据集预训练的特点，在ISCX nonVPN数据集上取得了不错的结果，但其计算开销非常大

Model	CIC-IoT		ISCX-Tor		ISCX-nonTor		ISCX-VPN		ISCX-nonVPN		Avg. Rank
	AC	F1	AC	F1	AC	F1	AC	F1	AC	F1	
AppScanner	0.9674	0.9620	0.7543	0.6163	0.9153	0.8273	0.8889	0.8722	0.7576	0.7486	6
K-FP	0.9349	0.9254	0.7771	0.6313	0.8741	0.8167	0.8713	0.8747	0.7551	0.7387	8
CUMUL	0.9153	0.9101	0.6686	0.4997	0.8605	0.7627	0.7661	0.7644	0.6187	0.5897	9
ETC-PS	0.9218	0.9122	0.7486	0.6033	0.9365	0.8486	0.8889	0.8851	0.7273	0.7208	7
FS-Net	0.9805	0.9759	0.8286	0.7242	0.9278	0.8285	0.9298	0.9234	0.7626	0.7555	5
DF	0.8664	0.8601	0.6514	0.4719	0.8568	0.7590	0.8012	0.7921	0.6742	0.6701	10
ET-BERT	0.9565	0.9122	0.9543	0.9397	0.9029	0.8332	0.9532	0.9463	**0.9167**	0.9235	4
GraphDApp	0.6808	0.7263	0.4286	0.2281	0.6936	0.5352	0.6491	0.5740	0.4495	0.3614	11
TFE-GNN	0.9699	0.9666	**0.9886**	0.9855	0.9390	0.8507	0.9591	0.9536	0.9040	**0.9240**	2
YaTC	0.9397	0.9105	0.9868	0.9869	**0.9579**	**0.9522**	0.9605	0.9671	0.7546	0.7544	3
MH-Net	**0.9900**	**0.9896**	**0.9886**	**0.9886**	0.9465	0.9453	**0.9942**	**0.9941**	0.9141	0.9141	**1**

实验设计

- 对比实验（包级对比实验）
  - MH-Net仍然比其他基线具有绝对优势，其次是EBSNN-LSTM和EBSNN-GRU
  - EBSNN在所有数据集上都显示出有竞争力的结果，但不能充分利用字节之间的信息相关性，从而导致性能瓶颈

Model	CIC-IoT		ISCX-Tor		ISCX-nonTor		ISCX-VPN		ISCX-nonVPN		Avg. Rank
	AC	F1	AC	F1	AC	F1	AC	F1	AC	F1	
Securitas-C4.5	0.8675	0.8571	0.9520	0.9429	0.8916	0.8652	0.7250	0.7250	0.5833	0.6212	4
Securitas-SVM	0.6526	0.7098	0.8229	0.8046	0.7333	0.7180	0.6375	0.6454	0.5750	0.6933	9
Securitas-Bayes	0.6667	0.7541	0.8083	0.6938	0.8062	0.7842	0.6125	0.6736	0.5416	0.6938	10
2D-CNN	0.7745	0.7863	0.3781	0.3576	0.6560	0.6877	0.2887	0.2219	0.5517	0.4638	11
3D-CNN	0.8788	0.8779	0.7837	0.7534	0.4791	0.4273	0.8445	0.8436	0.5346	0.5114	8
DP-SAE	0.7284	0.6134	0.8190	0.8078	0.8244	0.7889	0.7227	0.6849	0.6985	0.6952	6
DP-CNN	0.8664	0.8607	0.6622	0.6342	0.8784	0.8633	0.9270	0.9283	0.4326	0.3201	7
BLJAN	0.9643	0.9620	0.7580	0.7762	0.1348	0.2081	0.8539	0.7646	0.7356	0.7640	5
EBSNN-GRU	0.9440	0.9431	0.9319	0.9208	0.9271	0.9207	0.9467	0.9463	0.8103	0.8096	3
EBSNN-LSTM	0.9759	**0.9819**	0.7663	0.7183	**0.9519**	**0.9484**	0.9527	0.9531	0.8156	0.8128	2
MH-Net	**0.9806**	0.9800	**0.9916**	**0.9917**	0.9156	0.9122	**0.9768**	**0.9766**	**0.8822**	**0.8814**	**1**

实验设计

- 消融实验（ CIC IoT和ISCX-VPN数据集与F1分数）
  - 8位流量单元对模型性能的贡献大于4位流量单元，但后者所携带的信息也不容忽视
  - 将异构流量图转换为同构流量图后，结果显著下降，这表明有必要对流量单元之间的异构相关性进行建模
  - 对比学习的贡献：整合包级和流级对比损失可以提高分类性能，后者的影响更为显著

Variant	CIC-IoT		ISCX-VPN	
	Flow	Packet	Flow	Packet
w/o 4-bit View	0.9800	0.9681	0.9498	0.9355
w/o 8-bit View	0.8842	0.8970	0.1702	0.1488
w/o Hetero.	0.9154	0.9551	0.9120	0.9010
w/o L_{PCL}	0.9835	0.9618	0.9524	0.9524
w/o L_{FCL}	0.9767	0.9631	0.9467	0.9204
MH-Net	**0.9896**	**0.9800**	**0.9941**	**0.9766**

算法原理

- 算法总结
  - 算法贡献
    - 提出了一种名为MH-Net的模型，通过将不同数量的流量比特聚合到流量单元中，通过逐点互信息构建多视图异构流量图，采用异构图神经网络进行图表示学习
    - 以多任务方式进行监督对比学习，以获得更稳健的流量表示
  - 算法不足
    - 该模型在流量异构图的构建存在一定不足，没有整合三种不同的子图

特点总结与未来展望

研究工作总结

- 特点总结
  - TFE-GNN
    - 提出了一种构建字节级流量图的方法,挖掘原始字节之间的潜在相关性，设计TFE-GNN加密流量分类模型，从构建的流量图中提取高维特征
  - MH-Net
    - 根据不同长度的流量单元和不同位置流量单元的相关性构建多视角异构流量图，进一步挖掘流量单元之间的潜在相关性，以多任务方式进行监督对比学习，以获得更稳健的流量表示
- 未来展望
  - 不同层级流量特征的综合设计
  - 加密流量分类器的复杂度和实时处理能力

参考文献

- [1] Haozhen Zhang, Le Yu, Xi Xiao. TFE-GNN: A Temporal Fusion Encoder Using Graph Neural Networks for Fine-grained Encrypted Traffic Classification [C]. International World Wide Web Conference, 2023: 2066-2075.
- [2] Haozhen Zhang, Haodong Yue, Xi Xiao, et.al. Revolutionizing Encrypted Traffic Classification with MH-Net: A Multi-View Heterogeneous Graph Model [C]. Proceedings of the AAAI Conference on Artificial Intelligence, 2025, 39(1), 1048-1056.

知人者智，自知者明。胜人者有力，自胜者强。知足者富。强行者有志。不失其所者久。死而不亡者，寿。

谢谢！