分类号 _____ 编号_____

U D C_____ 密级_____

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

本科生毕业设计（论文）

题　　目：智能网关设备中的一种加密流量分类方法

姓　　名：许天淇

学　　号：11912921

系　　别：计算机科学与工程系

专　　业：计算机科学与技术

指导教师：宋轩 副教授

2023 年 6 月 2 日

诚信承诺书

1. 本人郑重承诺所呈交的毕业设计（论文），是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。

2. 除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。

3. 本人承诺在毕业论文（设计）选题和研究内容过程中没有抄袭他人研究成果和伪造相关数据等行为。

4. 在毕业论文（设计）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

作者签名: 潘天浦

2023 年 06 月 02 日

COMMITMENT OF HONESTY

1. I solemnly promise that the paper presented comes from my independent research work under my supervisor's supervision. All statistics and images are real and reliable.

2. Except for the annotated reference, the paper contents no other published work or achievement by person or group. All people making important contributions to the study of the paper have been indicated clearly in the paper.

3. I promise that I did not plagiarize other people's research achievement or forge related data in the process of designing topic and research content.

4. If there is violation of any intellectual property right, I will take legal responsibility myself.

Signature: 冯天慧

Date: 2023.06.02

智能网关设备中的
一种加密流量分类方法

许天淇

（计算机科学与工程系　指导教师：宋轩）

[摘要]：将集成数据中心的高空平台系统作为边缘数据中心在环保、提升用户体验等方面有很大的发展潜力。其中边缘数据中心的路由问题可以通过基于数据包级的加密流量分类模型的智能网关来解决。过去的基于深度学习的相关工作存在模型庞大，准确率低，开源代码质量低等问题。因此，在本文中，我们提出了一种数据包级的加密流量分类轻量级深度神经网络 Light Packet，其通过一维卷积神经网络和多头自注意力机制，在提高准确性的同时保持了较小的模型体积，在 ISCXVPN 2016 数据集上达到了 0.9749 的 F1 分数。此外，我们构建了一个功能丰富的数据包级流量分类的实验平台，方便未来相关方向的研究。

[关键词]：深度学习；加密流量分类；高空平台系统

[**ABSTRACT**]: The data center-enabled High-Altitude-Platform system as edge data center has great potential for development in terms of environmental protection and improving user experience. Among them, the routing problem of edge data centers can be solved by intelligent gateways based on packet-level encrypted traffic classification models. Past related work based on deep learning suffers from huge parameters, low accuracy, and low quality of open source code. Therefore, in this paper, we propose a packet-level encrypted traffic classification lightweight deep neural network called Light Packet, which achieves an F1 score of 0.9749 on the ISCXVPN 2016 dataset by improving accuracy while maintaining a small model size through the one-dimensional convolutional neural network and the multi-headed self-attentive mechanism. In addition, we construct a feature-rich experimental platform for packet-level traffic classification to facilitate future research in related directions.

[**Key words**]: Deep learning, Encrypted traffic classification, High Altitude Platform

目录

1. 引言

1.1 高空平台系统

高空平台系统，也可简写为 HAPS（High Altitude Platform System），是一种在 20 千米高度运行的计算机网络节点，通常充当无线通讯服务提供商的角色[1]。与卫星网络相比，由于其与地面用户距离较短且位置相对静止，高空平台系统为地面网络用户提供了较低延迟的连接，常用于为基础设施匮乏的偏远地区或在遭遇自然灾害的紧急情况下提供互联网连接服务[2]。

HAPS 的一个重要应用是集成数据中心功能。如今，数据中心行业的能源消耗呈指数级增长，使能源效率成为一个重要问题。服务器的冷却和供电是数据中心的主要能耗部分。集成数据中心的 HAPS 在低温（-50 摄氏度至-15 摄氏度）的平流层飞行，并配备大型太阳能电池板供电，大幅降低了能源成本[1,3]。集成数据中心的 HAPS 为用户提供了一个中等计算量任务的选择。

图1展示了一个将集成数据中心的 HAPS 作为边缘数据中心的网络拓扑结构。HAPS 充当边缘数据中心，与主数据中心共享工作负载。尽管主数据中心通常具有更大的存储容量和更多的计算资源，使其能够处理大型语言模型训练等高计算量的任务，但访问位于主干网络上的这些数据中心所造成的同样延迟不能被忽视。边缘计算被引入以解决主干数据中心的局限性。尽管边缘设备通常缺乏主数据中心那样大量的计算资源，但它们提供了较低的延迟。边缘计算将计算带到终端用户更近的地方，最大限度地减少数据传输距离，同时保持集中化。类似于智能手机等常用的边缘设备通常没有足够的计算资源来执行中等级别的深度学习算法。采用 HAPS 进行边缘计算既能提供低延迟的网络连接，又能提供高性能的计算资源，因此集成数据中心的 HAPS 作为边缘计算节点有很大的发展潜力。

边缘计算中的一个重要问题是确定哪些流量应该导向主数据中心，哪些应该导向边缘数据中心。我们在本研究中的目标是通过一个智能网关，自动决定传入流量的目的地，从而简化用户使用集成数据中心的 HAPS 的操作。目前，大多数数据包都使用 SSL（Secure Socket Layer）或 TLS（Transport Layer Security）加密，使运行在 IP 层以下的网关无法辨别传入流量的信息。因此，一个有前景的方法是采用一个加密网

络流量分类模型来完成流量的智能路由。该模型可以分析加密流量模式，并就流量应发送到主数据中心以执行更计算密集的任务还是发送到边缘数据中心以执行需要较低延迟的任务做出决策。实施这一解决方案将在保持安全性和数据完整性的同时，优化网络性能和用户体验。

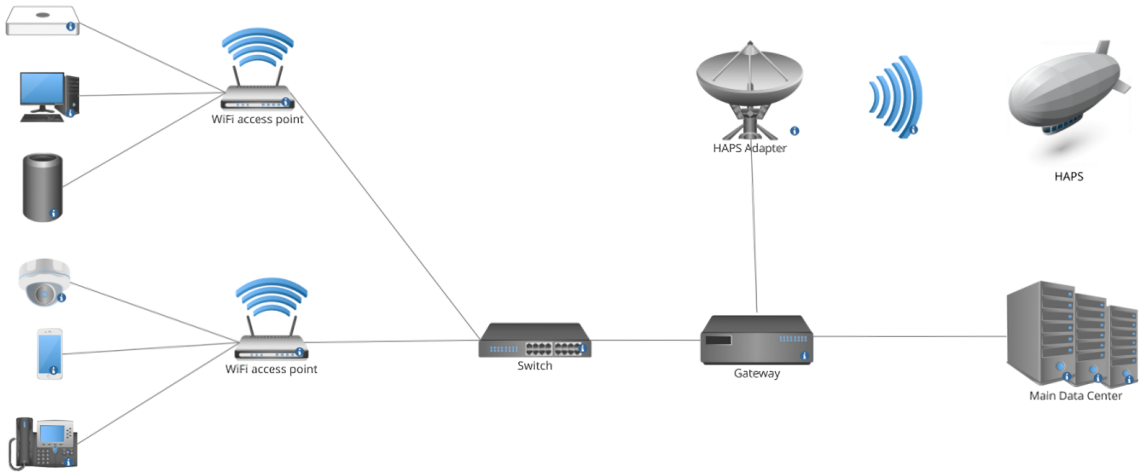

图 1 一种典型的集成数据中心的 HAPS 的网络拓扑结构

1.2 加密流量分类

网络流量分类和特征分析对于理解和解决基于互联网的应用中的各种问题越来越重要，如负载均衡、网络安全、边缘计算等。传统的流量分类通常使用基于端口的方法或 DPI（Deep Packet Inspection）方法。

基于端口的方法基于许多协议使用某个固定端口的现象。例如，HTTP 协议使用 80 端口，HTTPS 协议使用 443 端口，FTP 协议使用 22 端口等。然而，现在的网关设备常常使用随机端口来保护用户隐私或者采用 NAT 技术转换数据包暴露在外网的端口，而且大多数现代协议都没有固定的端口，这使得基于端口的方法失效。同时，随着 Web 技术的不断发展，许多性质上完全不同的流量都在 HTTP/HTTPS 协议上运行，包括文本流量、视频流量和图像流量。因此，为了优化用户体验，仅仅在协议层面分类是不够的，流量分类方法要尽可能地获取更多应用层的信息以更好地对网络进行优化。

为了解决基于端口方法的问题，DPI 方法被提出[4]。DPI 方法通过人工编程规则

的方式挖掘数据包中的模式或关键词，因此比基于端口的方法更可靠。然而，一些研究指出，DPI 在处理加密流量方面面临很大挑战[5]。如今，加密技术在互联网上得到了广泛应用，特别是 TLS 协议和 HTTPS 协议是现今 Web 技术的默认协议。这些为隐私保护而设计的加密技术成为了流量分类的障碍，因为人们很难从加密流量中提取模式和关键词。

在最近的学术研究中，基于机器学习的分类器受到了极大的关注。许多工作都是基于传统的机器学习算法，包括支持向量机、决策树和贝叶斯网络[6]。研究人员通常提取基于流的统计数据（如数据包大小、数据包持续时间和数据包数量）作为流量特征，并利用这些机器学习算法进行分类。然而，这些手工提取的特征需要不断改变以在不同分类任务上达成最好的效果，因此不存在通用的适用于每个任务的机器学习模型，这使得训练一个模型需要更多的专家知识和特征提取的时间。

在过去的十年里，基于深度神经网络的深度学习技术改变了世界。深度学习模型不仅在计算机视觉和自然语言处理的几乎所有任务中成为了最先进的技术，而且还对整个数据挖掘领域产生了巨大影响。深度学习在流量预测、空间数据处理和网络流量分析相关任务中也变得非常流行。深度学习的优势有很多方面。首先，深度学习模型是一种端到端的方法，这意味着它可以从以字节序列表示的原始数据包中自动提取特征，以实现特定任务。其次，由于深度学习模型通常学习具有区分性的特征，只要数据集具有相似的结构，它们就可以应用于各种任务。

值得一提的是，以前用于流量分类的大多数深度学习和机器学习模型都是基于流的，即输入是一个数据包序列。基于流模型的问题在于它们的高延迟，因此不适合用于数据中心的高速网络设备。此外，不同流量流的大小差异很大。如果我们在相同的层次上考虑它们，较大的延迟敏感流可能会阻塞较小的延迟不敏感流[7]。在这种情况下，用于细粒度网络管理的数据包级模型更有用，可以对每个独立的数据包进行分类。

尽管已经存在不少通过深度学习的方法进行数据包级流量分类的研究[7-9]，但这些研究通常存在几个致命问题。第一，不同的研究者对数据的预处理方式不同。如果进行不恰当地预处理容易导致模型在对应的数据集上过拟合从而影响泛化性。此外，

不同的预处理方法也使得不同方法准确率没法在一个体系下进行比较；第二，模型的参数量相对数据量不平衡。很多模型在参数量的选择上十分随意，常常出现参数量相对于数据量过多的现象。这不仅使得模型的体积增大，还会导致模型的推断速度下降，这些问题在对延迟极度敏感的网络设备中是不可接受的。第三，开源工作数量少且质量差。在数据包级分类模型的研究中，开源代码的工作极少。少数开源代码的工作或其他人对工作的复现的代码质量也很差，使得后续工作需要花费长时间复现代码。

1.3 核心贡献

为了解决上述问题，在本文中，我们提出了一种轻量级深度神经网络，用于在加密网络流量上进行数据包级在线流量分类，名为 Light Packet。在同等准确率下，该模型有较快的执行速度和较少的参数量，适合在资源量较少的网关设备中运行。此外，我们基于 PyTorch-Lightning 和 Hydra[10]构建了一个数据包级流量分类的实验平台，该平台由数据处理，网络结构，训练配置等相互解耦的模块构成，有超参数搜索，日志记录等便利特性。我们在该平台上实现了 Light Packet, Deep Packet[8], SAM[7]等模型和 ISCXVPN 2016 数据集[11]。得益于解耦的设计思路和文件化的配置方式，研究者不需要修改大量代码，而只需通过数行代码即可轻松向平台中加入新数据，新网络并进行实验。

2. 相关工作

2.1 基于机器学习的加密流量分类方法

根据上节陈述的事实，即基于端口的方法和 DPI 方法均不适用于加密流量的分类，本节仅关注基于机器学习和深度学习的加密流量分类方法。

基于机器学习的方法通常使用来自流量流的手工特征，例如端口、数据包大小和根据特定分类任务的数据包数量[7]。Lim, S 等人[12]在决策树、KNN（K-Nearest Neighbors）、支持向量机和朴素贝叶斯上测试不同类型的特征组。他们发现，在这些算法中，决策树的效果最好，并且基于熵的最小描述长度离散化在端口和数据包大小特征上显著提高了分类准确性。Gil, G 等人[11]设计了基于流的时间相关特征（如流的持续时间、每秒流字节和前向/后向到达间隔），并使用两种机器学习算法（C4.5 和 KNN）

来提高 VPN 流量分类的准确性。上述基于机器学习的方法需要专家对特定分类任务的知识来构建特征，且都是基于流的，这导致了算法的高延迟。

Wang, W 等人[13]是第一批将端到端深度学习方法应用于加密流量分类领域的研究人员。他们的模型输入是一个流或会话的前 784 字节。输入数据通过两个堆叠的一维卷积神经网络和最大池化层以及两个全连接层，最后得到一个类别数大小的向量，然后应用 softmax 函数得到概率。他们的结果显示，端到端方法可以轻松处理未加密数据，以实现高准确率，但在加密数据上效果不佳。基于 Wang, W 等人的工作，Lotfollahi, M 等人[8]开发了一种名为 Deep Packet 的模型，可以在数据包级别对加密流量进行分类，实现更高的准确率和更多的类别。然而，它的改进主要依赖于更大的输入维度和更多的全连接层，没有解决过拟合和模型运行速度缓慢的问题。该工作另一个致命问题是，数据预处理没有隐藏端口号，且其使用的 ISCX VPN-NonVPN 数据集存在端口号与对应类别一一对应的特征[14]。因此，其分类性能的优势很可能来源于对端口号的识别，但是不具有泛化性。

除了常用的一维卷积神经网络和全连接网络，许多研究者将目光投向了更先进的网络结构。比如在 Tranformer[15]中展现出惊人效果的注意力机制。Xie, G 等人[7]提出了一种用于数据包级在线分类的自注意方法，该方法应用自注意机制来提高分类模型的性能和可解释性。他们提出的模型在保证了分类准确性的基础上提高了推断速度，根据他们的测试，其对一个包的分类仅需 2 毫秒。在模型可解释上，他们发现，在数据包级别上，某几个字节位置对加密流量分类非常重要。然而，该研究缺乏消融实验，难以证明自注意机制对模型准确性和速度的贡献。尽管该研究开源了其实验代码，其代码质量较差，使得研究者难以在其基础上后续的研究。可解释性描述篇幅较短，内容较浅，没有结合源数据结构对模型在数据包加密情况下的分类能力做出进一步解释。

生成对抗网络（GAN）[16]也是深度学习中极其热门的模型，其有着极强的数据生成能力。Wang, P 等人[17]提出了一种名为 ByteSGAN 的用于加密流量分类的半监督生成对抗网络。通过利用 GAN 强大的数据生成能力，ByteGAN 可以充分利用少量标记的流量样本和许多未标记的样本，以实现良好的性能。由于其轻量级特性，它也非常

适合 SDN 边缘网关。

Aceto, G 等人[9]首次将多模态多任务深度学习应用于加密流量分类。他们全面分析和比较现有方法，并提出了一种名为 DISTILLER 的多模态多任务深度学习模型。通过学习内部和跨模态依赖关系，该模型利用流量数据的异质性，克服现有短视的单一模态深度学习流量分类的性能限制，并同时解决相关的不同流量分类问题。基于这项工作，Nascita, A 等人[18]通过可解释的人工智能技术研究信任度和可解释性，以理解、解释和改进 SOTA 流量分类器的行为，成功地提供了全局解释。

2.2 卷积神经网络

卷积神经网络（Convolutional Neural Network，简称 CNN）是一种广泛应用于计算机视觉、语音识别和自然语言处理等领域的深度学习模型。它们在处理图像、音频和文本数据时表现出了优越的性能，尤其擅长捕捉局部结构和模式。

卷积神经网络的主要组件包括以下几类层：

- 卷积层：卷积层使用一系列可学习的过滤器（也叫卷积核）来对输入数据进行卷积操作。这些过滤器可以捕捉图像的局部特征，如边缘、角点和纹理等。卷积操作可以保留空间信息，同时降低数据的维度。

- 归一化层：归一化层用于在神经网络中标准化输入或隐藏层的输出。归一化层有助于减少神经网络训练过程中的内部协变量移位问题，提高神经网络的性能。常用的归一化方法有批归一化（Batch Normalization）、层归一化（Layer Normalization）等。

- 激活函数层：激活函数层主要负责为网络引入非线性。常用的激活函数有 ReLU（Rectified Linear Unit）、tanh 和 sigmoid 等。非线性激活函数使得神经网络能够学习复杂的函数映射关系。

- 池化层：池化层是一种降采样操作，旨在减少卷积层输出的维度，从而降低计算复杂度。常见的池化方法有最大池化（Max Pooling）和平均池化（Average Pooling）。池化层可以提高模型的平移不变性，降低过拟合风险。

一维卷积神经网络（1D CNN）是卷积神经网络的一种变体，主要用于处理时序数据或序列数据，如时间序列分析、信号处理和自然语言处理等领域。与二维卷积神经网络（2D CNN）主要处理图像数据不同，1D CNN 主要关注数据在单一维度上的局部模式。而网络包数据作为一种以字节流的形式存储的序列数据，自然非常适合使用 1D CNN 进行处理。

1D CNN 的组成与一般的卷积神经网络类似，包括卷积层、激活函数层、池化层和全连接层。不过，它们在卷积操作和池化操作上有所不同，具体来说，他们的卷积（池化）核只在一维的方向移动，同时输出也是一维的。与全连接网络相比，1D CNN 通常具有较低的计算复杂度，因此可以在较长的序列数据上进行高效的训练和推理。另外，1D CNN 可以与其他神经网络结构（如循环神经网络、长短时记忆网络或 Transformer[15]）结合使用，以更好地捕捉序列数据中的长距离依赖关系。

2.3 多头自注意力机制

多头自注意力机制（Multi-Head Self-Attention）是一种用于处理序列数据的强大技术，尤其在 Transformer[15]架构中表现优异。假设 $X \in \mathbb{R}^{l \times dmodel}$ 为输入序列，其中 l 为序列长度，d_{model} 为序列中元素的嵌入维度。在多头自注意力中，输入序列的每个单元（如网络包中的字节）首先被映射到查询（Q）、键（K）和值（V）三个不同的空间。这些映射是通过训练过的权重矩阵来实现的，即 $Q = W_Q X$，$K = W_K X$，$V = W_V X$，其中 $W_Q, W_K, W_V \in \mathbb{R}^{d_{\text{model}} \times h d_k}$ 为权重矩阵，其中 d_k 表示键（K）的维度。接下来，查询、键和值矩阵被分割成多个头。每个头都有自己独立的权重矩阵，这使得模型能够学习到多种不同的注意力表示。我们将第 i 个头的查询、键、值矩阵分别记作 Q_i, K_i 和 V_i, 且他们的维度均为 $l \times d_k$.

然后，在每个头上分别计算缩放点积注意力，这是通过将查询矩阵与键矩阵的点积结果除以一个缩放因子（通常是键向量长度的平方根）并归一化得到的。接着，用归一化后的注意力分数对值矩阵进行加权求和，得到每个头的输出。之后，所有头的输出被拼接在一起，并通过一个线性层进行变换，以获得多头自注意力机制的最终输

出。这一过程可以被表示为：

$$\text{MultiHeadAttn} = \text{Concat}(\text{head}_1, \text{head}_2, ..., \text{head}_h)W_O \quad (1)$$
$$\text{where head}_i = softmax(\frac{Q_i K_i^T}{\sqrt{d_k}} V_i)$$

其中 h 代表头的个数，$W_O \in \mathbb{R}^{hd_k \times d_{\text{model}}}$ 为输出的权重矩阵。

多头自注意力机制的优势在于它能够并行处理输入序列中的所有单元，并学习多种不同的注意力表示，从而捕捉丰富的上下文信息。

3. 模型设计

结合一维卷积神经网络和多头自注意力机制，我们设计了 Light Packet 模型用于加密网络数据包分类。模型结构见图2。该模型主要包含两个模块，即由 1D CNN 构成的负载压缩网络（Payload Compress Net）和与 Transformer Encoder 结构相似的分类网络。在下面的章节中我们将详细介绍该网络的结构。

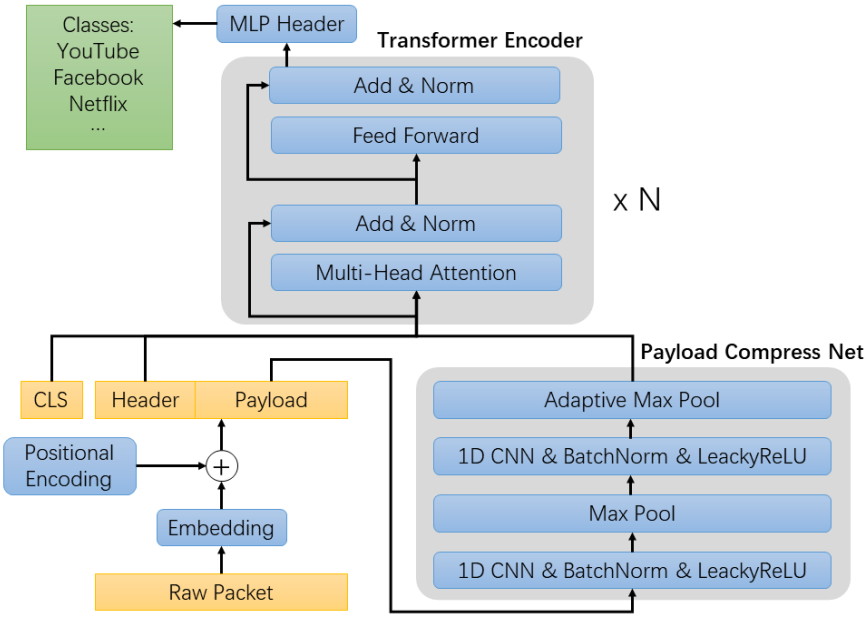

图 2 Light Packet 网络结构示意图

3.1 输入处理

原始的网络数据包是长度不固定的字节序列，为了将这些不规则的数据包转化为网络可以接受的形式，我们采取了两种策略。首先，如果数据包的长度超过我们设

定的固定长度，我们会从数据包的头部截取固定长度的字节序列；其次，如果数据包的长度不足，我们则通过填充补齐长度。这样，无论原始数据包的长度如何，我们都能得到一个固定长度的字节序列作为网络的输入，其中每个字节的取值都在 [0, 255] 的整数范围内。然后，我们将得到的固定长度字节序列输入到一个可学习的嵌入网络中，这个网络能够将每个字节转换为一个 d_{model} 维的向量。这些向量又被称为嵌入向量，它们会在网络训练过程中不断更新，以更好地表示输入字节的特征。

因为 Light Packet 中所用到的网络结构，包括 1D CNN，Multi-Head Attention 和全连接层，都是位置无关的，这意味着输入序列在这些网络中处理时会丢失其原始的位置信息。而在许多场景中，字节之间的顺序和关系是非常重要的，如果忽略这些信息，网络可能无法准确地处理数据。因此，我们引入了位置编码来解决这个问题。位置编码是通过向每个输入嵌入向量添加一个 d_{model} 维的位置向量来实现的，这个位置向量也是可以学习的，它能够在训练过程中不断更新以更好地表示位置信息。通过这种方式，我们将位置信息融入到了输入特征向量中，使得网络能够更好地处理序列中的顺序信息。

经过上述处理，我们得到了一个形状为 $l \times d_{\text{model}}$ 的序列嵌入，这个嵌入包含了位置信息。在这个嵌入中，我们根据原始数据包的格式，将序列划分为包头部分和负载部分。我们将负载部分输入到负载压缩网络中进行处理，然后将处理后的结果与包头部分在序列维度上进行拼接，得到的结果再输入到分类网络中进行处理。在这个过程中，我们根据 IP 包头和 TCP 包头的默认最小长度，将切分包头部分的长度设置为 40 字节。

3.2 负载压缩网络

从直觉上讲，网络包，特别是经过加密的网络包的负载，对于分类任务可能提供的信息相对较少。这是因为同一个应用的负载通常具有很高的随机性，包含的与应用分类直接相关的信息可能较少。更进一步，如果网络包在正确配置的加密后，负载部分应该不会包含任何对于分类任务有帮助的信息。鉴于此，我们提出了一个被称为负载压缩网络（Payload Compress Net）的概念。这个网络的主要目的是使用一维卷积神经网络来提取网络负载中的关键特征并压缩序列长度，从而减小后续分类网络的

计算负担，并提高其运行速度。

具体来说，负载压缩网络由两组卷积层和池化层构成。在卷积层中，我们选择了大小为 4 的卷积核，这是因为常用的互联网协议通常以 4 字节或 8 字节进行对齐，因此，选择大小为 4 的卷积核能够更好地提取每个字段的信息。此外，我们设定卷积层的 Stride 和 padding 都为 1，这样可以保证卷积层处理后的序列长度保持不变。同时，我们还将卷积层的通道数设定为 d_{model}，这样可以保证通道数也保持不变。在池化层方面，第一个池化层是一个包含大小为 4 的卷积核的最大池化层，选择这样的设计同样是为了更好地提取每个字段的信息。第二个池化层是一个自适应最大池化层，这个池化层能够保证输出序列的长度为一个确定的值。这个自适应池化层的存在使得我们能够控制保留的序列长度，从而有效地控制网络的大小。

在设计负载压缩网络时，我们之所以选择使用一维卷积神经网络，是因为一维卷积层的计算复杂度与序列长度成正比，而分类网络中使用的多头自注意力层的计算复杂度与序列长度的平方成正比。因此，通过负载压缩网络减小输入分类网络的序列长度，我们可以大幅度地降低分类网络的计算复杂度，从而有效地提高推理速度。数学上，对于输入输出序列长度均为 N，卷积核长度为 K，输入和输出通道数均为 d_{model}，步长等于 1 的卷积层，其时间复杂度可表示为

$$\mathbf{Time}_{\text{CNN}} \sim O(N \cdot K \cdot d_{\text{model}}^2) \tag{2}$$

而对于序列长度为查询（Q）、键（K）和值（V）维度均为 d_k，头数为 h 的多头自注意力层，其时间复杂度为

$$\mathbf{Time}_{\text{Attn}} \sim O(N^2 \cdot d_k \cdot h + 2N \cdot d_k^2 \cdot h) \tag{3}$$

由上式可得，对于序列长度 N，单层多头自注意力层的计算复杂度为 $O(N^2)$。因此，假设我们通过负载压缩网络将原本长度为 216 的负载压缩到长度为 4，而包头的长度为 40 时，多头注意力层的时间复杂度可以减少到原本的 3%。而引入负载压缩网络而增加的计算复杂度为 $O(N)$，相比于其降低的复杂度几乎可以忽略不记。

3.3 分类网络

在经过负载压缩网络后，我们将先前已嵌入的包头部分和压缩后的负载拼接，作为分类网络的输入。分类网络采用类似 Transformer Encoder 的结构。该分类网络由多层编码器层组成，每一层都有相同的结构。通常，根据任务的复杂性，可以设置不同的层数。每一层编码器包含两个主要子层：上文介绍的多头自注意力机制（Multi-Head Self-Attention）和位置前馈神经网络（Position-wise Feed-Forward Network）。多头自注意力机制能够计算输入序列中每个元素与其他元素之间的关系。这使得模型可以捕捉输入序列中远距离的依赖关系，而不仅仅是局部的信息。位置前馈神经网络则是一个逐位置的全连接层，它在不改变序列长度的情况下，为序列中的每个位置提供额外的非线性变换。这有助于模型捕捉更复杂的模式和关系。

每个子层都采用了残差连接和层归一化（Layer Normalization）技术。残差连接有助于减轻梯度消失问题，而层归一化则有助于加速训练过程和提高模型的稳定性和泛化能力。

另外，我们还参考了 BERT 模型[19]的设计，为输入序列的开头添加了一个 CLS 标记（Classification）。这个标记的主要作用是在分类模型进行自注意力操作和逐层编码后，CLS 标记所对应的输出向量将包含整个输入序列的全局信息。这样的设计使得我们可以将 CLS 标记的输出向量作为整个输入序列的汇总表示，这个汇总表示可以被输入到一个激活函数为 Softmax 的全连接层，得到最后的分类结果。这种方法使得我们的模型能够更好地理解输入序列的整体信息，从而提高分类的精度和效率。

4. 实验设计和结果

4.1 实验平台构建

为了准确、公平地比较模型性能，防止因数据预处理、模型测试方法、模型选择等问题带来的误差，我们通过 PyTorch-Lightning[1]（下文简称 PL）和 Hydra[10] 框架构建了一个数据包级流量分类的实验平台。该平台基于开源项目 lightning-hydra-template[2]，包含解耦的数据加载、网络结构、训练配置等相互解耦的模块。

[1]https://www.pytorchlightning.ai/index.html

[2]https://github.com/ashleve/lightning-hydra-template

数据加载功能是基于 PL 中的 LightningDataModule 模块实现的。该模块提供了数据加载过程的抽象，固定了数据拆分和数据准备流程，确保在不同的模型中保持一致。该模块可配置的参数包括输入序列长度、分类任务以及类别个数等。通过传入不同的数据集，同一个 DataModule 可以支持多个数据集，并保持相同的配置。我们在该模块中实现了两个分类任务，即应用分类和流量分类。

而网络结构功能是基于 PL 中的 LightningModule 模块实现的。该模块提供了模型和训练过程的抽象，固定了训练的优化目标、日志记录的评价指标以及训练、验证和测试过程中的不同操作。通过这个模块，用户可以通过修改配置文件的方式灵活配置所需的网络结构、优化器、学习率调度器等参数，而无需修改代码本身。

训练配置功能由 Hydra 实现，其优势在于能够通过编写和组合配置文件或命令行的方式动态创建分层配置，并进行覆盖修改或继承。Hydra 的配置文件基于易用的 YAML 格式，并且其参数可以直接注入 Python 函数中。将 Hydra 与上述 PL 模块相结合，用户即可灵活配置实验所用到的各参数，在不修改原有代码的情况下增加，删除和修改实验配置。

该实验平台还具有多样的日志记录、超参数搜索、模型测试等便利功能。通过该平台，研究者可以快速且灵活地实现和验证新想法。

4.2 数据集和预处理

ISCXVPN2016 数据集[11]是一种用于评估网络流量分类性能的公开数据集。该数据集特别关注于对 VPN 流量（Virtual Private Network，虚拟专用网络）和非 VPN 流量的分类。VPN 流量是通过 VPN 隧道传输的，以提高通信安全和保护用户隐私。非 VPN 流量是在没有 VPN 保护的情况下直接传输的。数据集中的流量类型涵盖了各种应用场景，如 Web 浏览、文件传输、视频流、音频流、邮件、P2P 以及 VoIP 等。这些流量类型在 VPN 和非 VPN 情况下都有涉及，从而为研究者提供了一个丰富的、多样化的用于评估流量分类算法的数据集。数据集中的数据包被捕获并保存在 PCAP 格式（Packet Capture）中，以便于研究者使用各种网络分析工具进行分析。PCAP 格式是网络分析领域常用的一种文件格式，可以储存原始的网络数据包。使用这种格式，研究者可以对网络数据包进行深入的分析，以从中提取特征并应用于分类模型。在这

个数据集中，捕获到的数据包被分隔到不同的 PCAP 文件中，并根据产生数据包的应用程序（如 Skype、Hangouts 等）以及在捕获会话期间应用程序参与的特定活动（如语音通话、聊天、文件传输或视频通话）进行标记。

我们参考了 Deep Packet[8]中对该数据集的预处理方法，并做出了一些改进。首先，因为 ISCXVPN2016 数据集包含链路层及以上的所有信息，我们去除了链路层包头，其主要包含的信息是对分类无意义的 MAC 地址以及一些纠错机制。在传输层中有两种常见协议，Transmission Control Protocol (TCP) 和 User Datagram Protocol (UDP)，因为他们的包头长度不同（即在没有额外选项的 TCP 的包头长度通常为 20 字节，而 UDP 为 8 字节），我们对每个 UDP 包的包头后加上 12 个字节的 0 字节填充，使 UDP 和 TCP 包的包头长度一样。这将方便神经网络分离传输层包头和负载，提升分类准确性。另外，数据中存在着大量的 Domain Name Service (DNS) 包，但是 DNS 的功能是做域名和 IP 地址间的转换，和我们的分类任务无关，因此我们也去除所有 DNS 包。

对于输入序列长度的选择，我们首先考虑到以太网的 MTU (Maximum Transmission Unit) 一般为 1500，即以太网最大包的长度。然而在真实网络环境中，特别是应用层被加密的情况下，传输层负载能提供的信息十分有限，且长度为 1500 的包多数是在大量的数据传输中处于整个 TCP 流中间位置的包，其被加密后的负载内容通常对分类毫无价值。因此我们设置序列长度为 256，使得其能包含全部包头信息作为主要分类方式，同时拥有一部分负载内容辅助分类器做出判断。

同时，对于每个数据包中都存在的 IP 地址和端口号，因为其内容和应用类型高度关联，而其值又取决于采集数据的机器。如果将其保留，分类器很可能直接学习 IP 地址和端口号与类别之间的关系，造成对于单一数据集的过拟合，极大地降低模型的泛化性。因此，我们遮盖了这部分内容。

对于数据集的标签处理，我们根据应用的不同对数据包打上对应标签，根据用户具体使用应用的不同分为 Facebook, ICQ, YouTube 等 15 个类别。其中每个分类下既有通常的应用访问流量，又有通过 VPN 访问的加密流量，使得分类难度大大增加。标签的具体内容见表1。

表 1 分类标签

序号	应用类别
0	AIM Chat
1	Email
2	Facebook
3	FTPS
4	Gmail
5	Hangouts
6	ICQ
7	Netflix
8	SCP
9	SFTP
10	Skype
11	Spotify
12	Vimeo
13	Voipbuster
14	Youtube

4.3 模型评价指标

在分类任务中，混淆矩阵（Confusion Matrix）是一种常用的评估方法，用于描述模型预测结果与实际标签之间的关系。混淆矩阵中的四个基本组成部分是真阳性（True Positive, TP）、假阳性（False Positive, FP）、真阴性（True Negative, TN）和假阴性（False Negative, FN）。真阳性是指模型将正类样本预测为正类的数量，假阳性是指模型将负类样本预测为正类的数量，真阴性是指模型将负类样本预测为负类的数量，假阴性是指模型将正类样本预测为负类的数量。

基于混淆矩阵，我们可以计算一系列评估指标，如精确率（Precision）、召回率（Recall）和 F1 分数（F1 Score）。精确率描述了在所有被预测为正类的样本中，实际为正类的样本所占的比例。其计算公式为：

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

召回率描述了在所有实际为正类的样本中，被预测为正类的样本所占的比例。其计算公式为：

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

F1 分数是精确率和召回率的调和平均值，它综合考虑了精确率与召回率的性能，以便在评估模型时能够兼顾查准率与查全率。其计算公式为：

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

在多类别分类任务中，每个类别都有各自的精确率，召回率和 F1 分数，需要通过平均或加权平均的方式得到一个模型的综合评价指标。在本实验中我们使用平均精确率，平均召回率和平均 F1 分数来作为模型的整体指标，使用混淆矩阵计算各类别的召回率来进一步判断模型在不同类别的分类性能。通过混淆矩阵和相关评估指标，我们可以全面了解分类模型的性能，从而更好地针对实际应用场景来优化模型。

为了全面评估我们的模型，除了模型的分类性能之外，我们还关注模型的参数量和运行速度。这两个指标在网络包分类的实际应用中同样非常重要，因为智能网络设备往往没有较多的存储和计算资源。为此，我们采用了 fvcore 库[3]提供的 Flop Counter 功能来对模型的参数量和运行速度进行测量。Flop Counter 是一种用于评估深度学习模型性能的工具。对于一个基于 PyTorch 框架的深度学习模型，Flop Counter 能够计算出模型的参数个数和浮点运算数（floating point operations）。参数个数是评价模型大小的重要指标，它直接决定了模型在内存或硬盘中占用的空间大小。而浮点运算数则是评估模型运行速度的主要指标。它表示了在进行一次前向传播（即模型推理）过程中，需要进行的浮点运算的次数。浮点运算数越多，模型进行推理的时间就越长。

4.4 基线选择

为了衡量 Light Packet 的性能，我们选取了 3 个前沿的基于深度学习的数据包分类模型：DP-SAE[8]，DP-CNN[8]和 SAM[7]作为基线。DP-SAE 由 5 个堆叠的全连接层构成。DP-CNN 由 2 个一维卷积层和 3 个全连接分类层构成。SAM 由自注意力层，以及并行的两个一维卷积层和全连接分类层构成。这三种基线模型的实现代码，我们都是从相应的论文的开源仓库中获取，并在此基础上进行了一些必要的修改，以适应我们的测试平台，保证评估过程的公平性。

[3]https://github.com/facebookresearch/fvcore

4.5 超参数和模型选择

为了尽可能公平地衡量每个模型的性能，我们使用 Optuna[20]工具对所有的基线模型和 Light Packet 进行超参数搜索。所用到的超参数搜索算法是 Tree-structured Parzen Estimator（TPE）[21]，一种用于高效搜索深度学习模型的超参数空间的贝叶斯优化方法。优化的超参数包括了学习率、卷积通道数、多头注意力头数、网络层数、Light Packet 中的负载压缩长度等对最终结果比较重要的超参数。优化目标为模型在验证集上的最高平均 F1 分数。对于每个模型，我们先随机超参数训练 10 次并评估它们的性能，接着基于已有的观测值对超参数进行 10 次优化和模型训练并评估其性能，取性能最好的一组结果作为该模型的最终结果。

4.6 训练过程

数据集以 8:1:1 的比例划分为训练集，验证集和测试集。测试集不会在训练过程和超参数搜索中被使用。对于所有模型我们均采用 Early Stop 模式，即训练 10 个 epoch 未提高验证集的最高 F1 分数就停止训练。

Light Packet 模型在一张 V100 上训练 20 个超参数搜索迭代，时间花费约为 12 小时。批大小为 256。单次训练时间约为 40 分钟，根据模型大小的不同有较大差异。最终得到的最优模型在训练集上训练了 17 个 epoch。为了探究 Light Packet 模型的潜力，我们也训练了名为 Light Packet+ 的增强模型。其与 LightPacket 模型的主要差异为，Light Packet+ 不为了控制模型的体积而限制一些关键参数的最大值，而是尽可能达到最好的效果。最终 Light Packet 系列的参数选择可见表2。其他基线模型的训练细节不在此详述。

表 2 超参数搜索结果

模型	d_{model}	channels	compress size	heads	classifier layers
Light Packet	128	128	4	2	2
Light Packet+	256	256	20	4	4

4.7 实验结果

Light Packet 系列模型和用于对比的基线模型的精确率、召回率、F1 分数和参数量如表3所示。其中精确率评估了模型的预测准确度，召回率评估了模型识别正例的

能力，F1 分数则是精确率和召回率的调和平均值，评价了模型的整体性能，而参数量和浮点运算数则反映了模型的复杂度和计算成本。值得注意的是，在本实验中参数量和浮点运算数是重要指标，因为在资源相对缺乏的网络设备中体积小，运行速度快的模型具有很大的优势。

表 3 实验结果

模型	精确率	召回率	F1 分数	参数量	浮点运算数
DP-SAE	0.9444	0.9444	0.9438	**309K**	**308K**
DP-CNN	0.9541	0.9601	0.9566	446K	1.33M
SAM	0.9699	0.9741	0.9718	861K	217M
Light Packet	0.9722	0.9746	0.9733	464K	20.8M
Light Packet+	**0.9737**	**0.9762**	**0.9749**	1.85M	143M

我们首先对比了三个基线模型 DP-SAE、DP-CNN 和 SAM，可以看到，参数量和模型整体表现大致呈现正相关。在 Deep Packet 的两个模型 DP-SAE 和 DP-CNN 中，尽管 DP-CNN 的参数量略高，但其精确率、召回率和 F1 分数均超过了 DP-SAE，表明一维卷积层使得模型在该分类任务中相比于简单的全连接层堆叠有很大提升。而 SAM 模型将朴素的自注意力与一维卷积神经网络结合，在 DP-CNN 的基础上大幅提高了模型表现，使 F1 分数达到了 0.97 以上。然而，自注意力机制同时引入了更多参数和巨大的运算开销，其参数量大约为 DP-CNN 的两倍，而其计算量甚至达到了 DP-CNN 的 160 倍。

接着，我们对比了 Light Packet 系列模型和基线模型，结果显示 Light Packet 模型的参数量为 SAM 模型的一半，且计算量仅有 SAM 模型的 0.1，但却达到了高于 SAM 模型的精确率、召回率和 F1 分数，说明了 Light Packet 模型在该分类任务中的优越性。这得益于负载压缩网络在降低模型计算复杂度上的重要作用。作为 Light Packet 的加强版本，Light Packet+ 模型在所有指标上均达到了最高值。尽管其参数量较大，计算量与 SAM 模型也不相上下，但该实验结果表明 Light Packet 结构有很强的泛化性，在参数量提高的同时模型性能也在稳步提高，这得益于 Tranformer Encoder 结构的强大特征提取能力和泛化能力。

此外，我们还通过混淆矩阵计算了模型在各类别上的召回率，即正确分类个数在

该类别总数中所占比例，见表4。通过观察 Light Packet 和 Light Packet+ 在各类别上的召回率，我们发现 Light Packet+ 在大多数类别上相比 Light Packet 有所提升，这进一步证实了其性能的优越性。然而，两个模型在 Facebook、Hangouts 和 Skype 这三个应用上的性能都相对较差。这可能是由于这三个应用都是社交媒体软件，其网络数据包含文字、音频流、视频流等多种模态，导致了应用内部的数据模态差异大，而不同应用间的数据分布差异小，这增加了分类的难度。这也为我们进一步改进模型提供了方向，即需要考虑如何处理在网络包中的多模态数据，并提高模型在这类应用上的性能。

表 4 各类别召回率

应用类别	DP-CNN	SAM	Light Packet	Light Packet+
AIM Chat	0.91	0.96	0.95	0.95
Email	0.98	0.99	0.99	0.99
Facebook	0.93	0.94	0.93	0.94
FTPS	1.00	1.00	1.00	1.00
Gmail	0.94	0.99	0.98	0.98
Hangouts	0.93	0.94	0.94	0.94
ICQ	0.90	0.93	0.96	0.97
Netflix	0.99	1.00	1.00	1.00
SCP	0.95	0.97	0.96	0.97
SFTP	0.99	1.00	1.00	1.00
Skype	0.89	0.92	0.92	0.93
Spotify	0.99	1.00	1.00	0.99
Vimeo	1.00	1.00	1.00	1.00
Voipbuster	0.99	1.00	1.00	1.00
Youtube	0.99	1.00	1.00	1.00

5. 结论

本文提出了一种轻量级的基于一维卷积神经网络和多头自注意力机制的深度神经网络 Light Packet。该模型的新颖之处在于首次提出了应用于数据包分类的负载压缩网络结构，并且将 Transformer Encoder 结构运用于数据包分类任务。该模型能够在较少地参数量下获得更高的性能，同时增加该模型的参数量可以显著地提高模型性能。其轻量级的特性有利于在网络设备中的部署，例如集成数据中心的高空平台系统。为了将该模型与其他基线模型对比，也为了放便未来的研究者，我们构建了一个功能丰富的数据包级流量分类的实验平台并实现了 Light Packet 模型和多个基线模

型。我们在该平台上通过超参数搜索准确地测试了 Light Packet 和其他基线模型的性能，得到的结果佐证了 Light Packet 模型的优越性。最后，本文通过混淆矩阵分析了 Light Packet 模型在该数据集和分类任务上的表现，得出了其在部分类中分类性能不佳的原因。

未来的研究者可以通过我们提出的测试平台上进一步探究网络流量数据包级别的分类任务的算法。另外，加密流量包的分类模型可解释性依然有很高的研究空间和研究价值，其研究结果将会影响下一代网络加密协议和算法的设计。

参考文献

[1] KARABULUT KURT G, KHOSHKHOLGH M G, ALFATTANI S, et al. A Vision and Framework for the High Altitude Platform Station (HAPS) Networks of the Future[J]. IEEE Communications Surveys & Tutorials, 2021, 23(2): 729-779. DOI: 10.1109/COMST.2021.3066905.

[2] HOSHINO K, SUDO S, OHTA Y. A Study on Antenna Beamforming Method Considering Movement of Solar Plane in HAPS System[C]//2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall). 2019: 1-5. DOI: 10.1109/VTCFall.2019.8891546.

[3] MERSHAD K, DAHROUJ H, SARIEDDEEN H, et al. Cloud-Enabled High-Altitude Platform Systems: Challenges and Opportunities[J]. Frontiers in Communications and Networks, 2021, 2.

[4] ALCOCK S, NELSON R. Libprotoident: Traffic Classification Using Lightweight Packet Inspection[J]., 2012.

[5] DAINOTTI A, PESCAPÉ A, CLAFFY K. Issues and Future Directions in Traffic Classification [J]. IEEE Network, 2012, 26(1): 35-40. DOI: 10.1109/MNET.2012.6135854.

[6] AULD T, MOORE A, GULL S. Bayesian Neural Networks for Internet Traffic Classification[J]. IEEE Transactions on Neural Networks, 2007, 18(1): 223-239. DOI: 10.1109/TNN.2006.883010.

[7] XIE G, LI Q, JIANG Y. Self-Attentive Deep Learning Method for Online Traffic Classification and Its Interpretability[J]. Computer Networks, 2021, 196: 108267. DOI: 10.1016/j.comnet.2021.108267.

[8] LOTFOLLAHI M, JAFARI SIAVOSHANI M, SHIRALI HOSSEIN ZADE R, et al. Deep Packet: A Novel Approach for Encrypted Traffic Classification Using Deep Learning[J]. Soft Computing, 2020, 24(3): 1999-2012. DOI: 10.1007/s00500-019-04030-2.

[9] ACETO G, CIUONZO D, MONTIERI A, et al. DISTILLER: Encrypted Traffic Classification via Multimodal Multitask Deep Learning[J]. Journal of Network and Computer Applications, 2021, 183–184: 102985. DOI: 10.1016/j.jnca.2021.102985.

[10] YADAN O. Hydra - A framework for elegantly configuring complex applications[EB/OL]. 2019. https://github.com/facebookresearch/hydra.

[11] DRAPER GIL G, HABIBI LASHKARI A, MAMUN M, et al. Characterization of Encrypted and VPN Traffic Using Time-Related Features[C]//. 2016. DOI: 10.5220/0005740704070414.

[12] LIM Y S, KIM H C, JEONG J, et al. Internet Traffic Classification Demystified: On the Sources of the Discriminative Power[C]//Proceedings of the 6th International Conference on Emerging Networking Experiments and Technologies, Co-NEXT'10. 2010: 9. DOI: 10.1145/1921168.1921180.

[13] WANG W, ZHU M, WANG J, et al. End-to-End Encrypted Traffic Classification with One-Dimensional Convolution Neural Networks[C]//2017 IEEE International Conference on Intelligence and Security Informatics (ISI). 2017: 43-48. DOI: 10.1109/ISI.2017.8004872.

[14] PETER F. Analysis of the ISCX VPN-nonVPN Dataset 2016 for Encrypted Network Traffic Classification[EB/OL]. 2019. https://github.com/Mr-Pepe/iscx-analysis/blob/master/report.pdf.

[15] VASWANI A, SHAZEER N, PARMAR N, et al. Attention Is All You Need[Z]. 2017. arXiv: 1706.03762 [cs] [2023-02-06]. DOI: 10.48550/arXiv.1706.03762.

[16] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative Adversarial Nets[C]// Advances in Neural Information Processing Systems: vol. 27. Curran Associates, Inc., 2014 [2023-04-09].

[17] WANG P, WANG Z, YE F, et al. ByteSGAN: A Semi-Supervised Generative Adversarial Network for Encrypted Traffic Classification in SDN Edge Gateway[J]. Computer Networks, 2021, 200: 108535. DOI: 10.1016/j.comnet.2021.108535.

[18] NASCITA A, MONTIERI A, ACETO G, et al. XAI Meets Mobile Traffic Classification: Understanding and Improving Multimodal Deep Learning Architectures[J]. IEEE Transactions on Network and Service Management, 2021, 18(4): 4225-4246. DOI: 10.1109/TNSM.2021.3098157.

[19] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[C/OL]//. 2018. https://arxiv.org/abs/1810.04805.

[20] AKIBA T, SANO S, YANASE T, et al. Optuna: A Next-generation Hyperparameter Optimization Framework[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2019.

[21] BERGSTRA J, BARDENET R, BENGIO Y, et al. Algorithms for Hyper-Parameter Optimization [C/OL]//SHAWE-TAYLOR J, ZEMEL R, BARTLETT P, et al. Advances in Neural Information Processing Systems: vol. 24. Curran Associates, Inc., 2011. https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

致谢

首先，我要感谢我的导师宋轩教授和 KAUST，联合导师 Basem Shihada 教授和论文评阅老师唐博教授。在毕业论文的撰写过程中，教授们给予了我耐心的指导、建设性的建议和宝贵的支持。我也要感谢宋轩实验室的赵奕辰老师，他全程对接论文进度，是我的毕业论文能够顺利完成的保证。

同时，我要感谢我的同学们和朋友们，他们在学术上和生活上给予了我许多帮助和鼓励。特别感谢王一帆，黄北辰，金肇轩和魏宇同学，在我海外交流的过程中，他们帮我处理各种校内事务，也帮助我排解海外生活和学习上的压力，支持我走过忙碌的大四学年。

最后，我要感谢天才美少女帕拉瑞斯小姐。我抱着对她的憧憬与仰慕一步步走到今天。她是我心灵的支柱，我的光，我的月亮。