中国科学院大学

University of Chinese Academy of Sciences

硕士学位论文

轻量化流量识别技术研究

作者姓名：赵浩宇

指导教师：于爱民 正高级工程师

中国科学院信息工程研究所

学位类别：工程硕士

学科专业：网络信息与安全

培养单位：中国科学院信息工程研究所

2023 年 6 月

‘‘‘

Research on Interpretability Techniques for Traffic Identification Based on Deep Learning

A thesis submitted to

University of Chinese Academy of Sciences

in partial fulfillment of the requirement

for the degree of

Master of Network and Information Security

in Network and Information Security

By

ZHAO Haoyu

Supervisor: Professor YU Aimin

Institute of Information, Engineering, CAS

June, 2024

中国科学院大学
学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明或致谢。

作者签名：

日　　期：

中国科学院大学
学位论文授权使用声明

本人完全了解并同意遵守中国科学院有关保存和使用学位论文的规定，即中国科学院有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：　　导师签名：

日　　期：　　日　　期：

摘 要

流量识别在维护网络安全和管理网络资源方面扮演着至关重要的角色。通过有效识别网络流量，可以帮助识别并阻止恶意活动，优化网络资源分配，并提高网络效率。在此背景下，深度学习流量识别方法的定义和应用变得尤为重要。深度学习流量识别是指使用深度学习技术自动分析和分类网络流量，这种方法能够处理复杂的数据集，提取有意义的特征，从而更准确地识别和分类流量。但它也存在一些缺点。其中最主要的是对硬件资源的高需求。深度学习模型，尤其是大型模型和复杂的神经网络，需要大量的计算能力和存储空间。这些需求在硬件资源有限的环境中成为了一个显著的限制因素，尤其是在处理大规模数据或需要实时处理的应用场景中。鉴于这种情况，研究和开发轻量化的深度学习模型以进行流量识别变得至关重要。轻量化模型旨在减少对计算资源的需求，同时保持较高的准确性和效率。通过简化模型结构、减少参数数量和优化计算流程，轻量化模型能够在有限的硬件资源条件下运行，同时减少训练和推理时间，提高整体性能。因此，在资源受限的环境中，轻量化模型为深度学习流量识别提供了一种可行且高效的解决方案，有助于在保证网络安全和效率的同时，降低硬件资源的使用和成本。

本论文主要聚焦于研究在硬件资源（存储、算力、数据集大小）受限的条件下，使用深度学习模型进行网络流量识别的方法。现有的流量识别方法在处理应用流量、高维数据和大规模数据集时面临着存储空间不足、计算资源过载以及数据采集效率低下的问题。针对这些问题，我们提出了一种综合性的优化方法。对于存储空间的限制问题，我们通过分析模型的可解释性，计算不同特征对模型分类决策的贡献度，识别出模型分类中重点依赖的特征序列。这一方法不仅在不牺牲分类准确率的前提下减少了存储空间的占用，而且在数据采集阶段实现了数据的有效压缩，仅收集对模型预测结果至关重要的信息。针对计算资源的优化，我们采用了轻量化模型的方法，通过减少模型层数和简化循环结构，有效减少了矩阵运算和梯度计算，降低了模型收敛的迭代次数，从而加速了模型的训练过程。本文具体包含以下创新点：

研究点一：为了在保持分类准确率的同时节省存储空间，我们通过分析**模**

型的可解释性来计算不同特征对于模型分类决策的贡献度。这一过程使我们能够识别模型重点依赖的特征序列，进而指导我们对原始数据集进行**精细化采集**，从而得到一个远小于原始数据集的新数据集。这种方法不仅维持了模型的分类性能，同时显著降低了对存储资源的需求。通过这种方式，我们能够在数据采集阶段就实现数据的有效压缩，确保只收集对模型预测结果至关重要的信息，从而在资源利用上实现了优化。

研究点二：在研究一通过模型可解释性的研究和精细化数据采集基础上，为了进一步提升模型训练的效率和减少训练时间，我们提出了采用**轻量化模型**的策略。具体来说，我们通过对模型结构的仔细分析和优化，力求在保持准确率不变的前提下减少模型的复杂度。这一过程包括使用**知识蒸馏**技术，设计一个轻量级的模型作为原模型的 student 模型，通过减少模型的层数以及简化每层的循环结构来简化模型节省。这样的结构简化直接减少了矩阵运算和梯度计算的数量，从而降低了模型收敛到最优解所需的迭代次数。这一改进显著加快了模型的训练速度，提高了训练过程的整体效率。通过这种方式，我们不仅优化了模型的存储和数据采集，也加快了模型训练过程，实现了从数据采集到模型训练的全流程优化。

关键词：深度学习，流量识别，可解释性，知识蒸馏，模型解释

Abstract

Traffic identification plays a crucial role in maintaining network security and managing network resources. Effective identification of network traffic helps in detecting and preventing malicious activities, optimizing the allocation of network resources, and enhancing network efficiency. In this context, the definition and application of deep learning methods for traffic identification become particularly important. Deep learning traffic identification refers to the automatic analysis and classification of network traffic using deep learning techniques. This method can handle complex datasets and extract meaningful features, thus enabling more accurate identification and categorization of traffic. However, it also has some drawbacks, the most significant being its high demand for hardware resources. Deep learning models, especially large models and complex neural networks, require substantial computational power and storage space. These demands become a significant limiting factor in environments with limited hardware resources, especially when dealing with large-scale data or applications requiring real-time processing. Given this situation, researching and developing lightweight deep learning models for traffic identification becomes crucial. Lightweight models aim to reduce the demand for computational resources while maintaining high accuracy and efficiency. By simplifying the model structure, reducing the number of parameters, and optimizing the computational process, lightweight models can operate under limited hardware resource conditions, reducing training and inference time and improving overall performance. Therefore, in resource-constrained environments, lightweight models provide a viable and efficient solution for deep learning-based traffic identification, helping to ensure network security and efficiency while reducing the use and cost of hardware resources.

This paper focuses on the use of deep learning models for network traffic identification under conditions of limited hardware resources (storage, computing power, network data). Existing traffic identification methods face challenges of insufficient storage space, computational resource overload, and low efficiency in data collection

when dealing with application traffic, high-dimensional data, and large datasets. To address these issues, we propose a comprehensive optimization method. For the problem of limited storage space, we analyze the model's interpretability, calculating the contribution of different features to the model's classification decision and identifying the sequence of features that the model heavily relies on. This method not only reduces the use of storage space without sacrificing classification accuracy but also achieves effective data compression during the data collection phase, ensuring only the collection of information crucial to the model's prediction results. For computational resource optimization, we adopt a lightweight model approach, reducing the number of model layers and simplifying the loop structure, effectively reducing matrix operations and gradient calculations, thereby lowering the number of iterations needed for the model to converge to the optimal solution, and speeding up the training process. This paper contains the following innovative points:

Research Point One: To save storage space while maintaining classification accuracy, we analyze **the model's interpretability** to calculate the contribution of different features to the model's classification decisions. This process allows us to identify the sequence of features the model heavily relies on, guiding us to perform refined data collection on the original dataset, resulting in a new dataset significantly smaller than the original. This method not only maintains the model's classification performance but also significantly reduces the demand for storage resources. In this way, we achieve effective data compression during the data collection phase, ensuring only the collection of information crucial to the model's prediction results, thereby optimizing resource utilization.

Research Point Two: Building on our previous research on model interpretability and refined data collection, which already achieved significant optimization in storage space, we further proposed the strategy of using **lightweight models** to enhance model training efficiency and reduce training time. Specifically, we carefully analyzed and optimized the model structure to reduce its complexity while maintaining accuracy. This process included reducing the number of model layers and simplifying the structure of each layer. Such structural simplification directly reduced the number of matrix opera-

tions and gradient calculations, thereby decreasing the number of iterations required for the model to converge to the optimal solution. This improvement significantly sped up the model's training process and enhanced the overall efficiency of the training. In this way, we not only optimized the model's storage and data collection but also accelerated the model training process, achieving optimization throughout the entire process from data collection to model training.

Keywords: Deep Learning, Traffic Identification, Interpretability , Model Explanation

目 录

图形列表

表格列表

第 1 章　绪论

1.1　研究背景

随着互联网的迅速发展和网络应用的日益丰富，网络流量的管理和分析成为了网络安全和网络管理中的一个重要课题。网络流量识别，即通过分析网络数据流来识别和分类不同类型的网络活动，是保证网络安全和高效网络管理的关键。

在传统的网络环境中，流量识别主要依赖于端口号和特定的协议特征。然而，随着网络技术的发展，这些方法面临诸多挑战。例如，动态端口分配、加密通信协议的广泛应用，以及复杂多变的网络应用场景使得传统方法逐渐失效。此外，网络攻击和恶意软件的不断演化也增加了流量识别的难度。最初，网络流量识别主要依赖于静态特征，如端口号和协议类型。这些方法的基础是相对简单的规则和模式匹配技术。基于端口的识别方法依赖于预定义的端口号来识别特定的应用程序，基于协议的识别方法根据网络协议的特定特征来识别流量。这些方法在早期网络环境中相对有效，但随着网络应用的发展，它们开始面临诸多挑战。首先是动态端口使得基于端口号的识别方法不再可靠，其次是加密协议（如SSL/TLS）的普及使得流量内容难以被检测和分析，还有新的应用程序采用自定义或混淆的协议，增加了识别难度。为了克服这些挑战，研究者开始探索使用机器学习方法来识别和分类网络流量。这些方法不再依赖于静态特征，而是通过分析流量的统计特征和行为模式来识别。研究者从流量中提取统计特征，如数据包大小、时间间隔等特征，然后使用机器学习算法（如支持向量机、决策树）对流量进行分类。随着深度学习技术的发展，其在流量识别领域的应用日益增加。深度学习模型能够处理更复杂的数据并自动提取关键特征。举例来说卷积神经网络（CNN、VGGNet、ResNet）用于提取时间和空间特征，循环神经网络（RNN、LSTM、）适合处理序列数据，如时间序列流量。

深度学习技术在网络流量分析领域的应用已经显示出显著的优势，尤其是在特征提取、数据处理能力、准确性和灵活性方面。首先，深度学习模型，如卷积神经网络（CNN）和循环神经网络（RNN），能够自动从大量网络数据中提取复杂特征，从而减少了传统机器学习方法中对手动特征工程的依赖。这种自动化

的特征学习能力使得深度学习模型能够有效地识别网络流量中的复杂模式，即使这些模式对传统方法来说可能过于微妙或复杂。WiKibook Bibliography其次，深度学习模型的扩展性和实时处理能力使其非常适合处理网络环境中产生的海量数据。这些模型不仅可以有效处理大规模数据集，还能够在接近实时的环境中进行分析，对于需要快速反应的系统（如入侵检测系统）尤为重要WiKibook Bibliography。此外，深度学习模型在多项研究中表现出的高准确率和对数据中小变化和噪声的鲁棒性，使其在动态和嘈杂的网络环境中表现出色。最后，深度学习模型的灵活性和适应性也是其在网络流量分析中备受推崇的原因。通过技术如迁移学习，这些模型可以适应不同类型的网络环境和流量模式，并且可以根据特定的网络配置和安全需求进行定制和优化。在线学习能力和模型更新机制进一步提高了这些模型对新的流量模式和安全威胁的适应能力。WiKibook Bibliography 综上所述，深度学习在网络流量分析中的应用提供了强大的工具，能够提高系统的效率、准确性和适应性，从而成为提升网络安全和性能分析的关键技术。

人工智能的到来在流量识别技术中起到了显著的效果，但是也为其带来了新的挑战。首先是数据集的规模日益增长，导致了巨大的存储压力。网络流量的数据量庞大，且随着网络活动的增加持续增长，这不仅要求有足够的存储空间来保存这些数据，还需要高效的数据处理和访问机制以支持实时或近实时的流量分析。此外，随着模型复杂度的增加，模型中的参数数量也随之增多，这直接导致了训练时间的显著增长。复杂的模型虽然在处理多变的网络环境和识别精度方面具有优势，但同时也需要更多的计算资源和时间来训练，特别是在面对大规模数据时。这不仅增加了计算成本，也对快速部署和更新模型带来了挑战。因此，研究者和工程师们需要在模型的复杂性、训练效率和实际应用需求之间寻找一个平衡点。例如，通过模型优化、轻量化技术或者采用更高效的训练策略来减少模型的训练时间和计算需求。同时，有效的数据管理和处理策略也是解决存储压力的关键，比如使用数据压缩、选择性采样或者分布式存储系统等方法。这些挑战要求不断创新和改进现有技术，以适应网络流量识别领域的快速发展和日益增长的需求。因此，开发一个轻量化的流量识别模型是当前研究的一个重要方向。

1.1.1　现有问题

针对当前的流量识别研究中，我们面临一个显著的挑战：随着数据集变得越来越庞大和神经网络模型（例如当前的大型模型）变得越来越复杂，对内存、存储空间和计算能力等硬件资源的需求也不断增加。这一趋势在实际工程应用中造成了显著的挑战，特别是在硬件资源受限的情况下。对此，我们分别从数据集优化和模型优化两个方面来探索解决方案，并对不同的方法的优缺点进行阐述，最后概括出需要解决的具体问题。

针对数据集优化方面，不同处理方式对应着不同的影响。在当前的数据驱动时代，我们正面临着数据规模的快速增长，尤其显著的是在网络流量识别和其他大数据应用领域。目前，研究和商业应用中处理的数据集规模（如表 1.1 所示）已经达到了 TB 甚至 PB 级别，这种海量数据的涌现带来了一系列挑战，尤其体现在数据存储、处理效率和计算资源方面。庞大的数据集对硬件资源的巨大需求主要体现在存储空间和计算能力方面。处理和分析大数据集需要大量的内存和高性能的处理器，这对于资源有限的环境是一个重大挑战。同时，大数据集在数据处理和模型训练方面的效率问题也不容忽视。在训练深度学习模型时，需要处理更多的信息，这不仅增加了训练时间，还可能导致算法优化过程中的困难。此外，大规模数据集的分析和处理也意味着更高的能耗和更大的环境影响，这在当前对可持续发展日益关注的背景下尤为重要。为应对这些挑战，研究和工业界提出了多种解决方案。云存储和分布式文件系统的使用成为了主流，它们提供了可扩展、灵活且成本效益高的数据存储解决方案。然而，数据的访问和传输速度受网络带宽和稳定性的限制，而存储在云端的数据可能面临安全风险和隐私泄露的问题。高效的数据索引和查询技术可以加快对大数据集的访问和处理速度，但这需要精心设计以适应特定的数据结构和需求，且构建和维护索引可能需要额外的计算和存储资源。在数据处理和模型训练方面，对数据进行下采样或选择性采样是一种有效的策略，它可以只处理数据集中最重要或最具代表性的部分。这种方法降低了对计算资源的需求，加快了处理速度，但如果采样不当，可能会导致分析结果的偏差，甚至可能丢失数据集中的一些重要信息。此外，分布式计算框架如 Apache Hadoop 和 Spark 能够有效地利用集群的计算资源来处理和分析大数据集。这些框架通过并行处理和将数据集分割成小块，大大加快了数据处理的速度，并提高了容错性。然而，集群的运行需要大量的计算资源和电力，且设

置与维护成本高，需要专业知识来部署和维护。综上所述，虽然这些解决方案有效地应对了大数据带来的挑战，但它们也带来了新的问题和考虑因素。在选择具体方案时，需要综合考虑数据特性、业务需求、成本和资源限制等因素，以找到最适合的解决方案。相比上述谈论到的处理方法，我们需要进一步的探索其他数据集优化的方法。

表 1.1 神经网络模型的参数数量和训练时间表

Table 1.1 table of neural network model parameters and training time

神经网络模型	参数数量（大约）	最久训练时间（估计）
LeNet-5	60,000	数小时
AlexNet	60 million	5-6 天
VGG-16	138 million	2-3 周
Inception v3	23.8 million	1-2 周
ResNet-50	25.6 million	1-2 周
ResNet-101	44.5 million	2-3 周
ResNet-152	60 million	3-4 周
DenseNet-121	8 million	1 周左右
DenseNet-201	20 million	2 周左右
Transformer	65 million	数天到 1 周
BERT Base	110 million	4-5 天
BERT Large	340 million	1 周左右
GPT-2 Small	124 million	1-2 周
GPT-2 Medium	355 million	2-3 周
GPT-2 Large	774 million	3-4 周
GPT-2 XL	1.5 billion	4-5 周
GPT-3	175 billion	数月

针对模型优化的问题，在近年来的深度学习领域，神经网络的规模已经达到了前所未有的复杂度。例如，大型网络如 GPT-3 拥有超过 1750 亿个参数，而 BERT 大型模型也有数亿个参数（如表 1.1 所示）。这些网络通常包含成百上千个层次，每个层次都由大量的神经元和连接构成。如此庞大和复杂的神经网络带来了显著的性能提升，但同时也引发了一系列问题。首先，大型网络的训练成本极高。这不仅包括计算资源的成本，如使用大量 GPU 或 TPU 资源，也包括时间

成本，因为训练这样的模型可能需要数周甚至数月的时间。此外，巨大的模型规模也对存储空间提出了更高的要求。这些因素使得大型网络的训练和部署变得不易被普通用户或小型企业所承担。其次，越来越长的训练时间会带来一系列后果。例如，模型的迭代速度变慢，这可能会影响研究和产品开发的效率。长时间的训练还可能导致能源消耗过大，引发环境和经济上的考虑。为了解决这些问题，研究者们正在探索各种模型优化策略。模型剪枝技术通过移除神经网络中不重要的连接或神经元，有效减少模型的大小和复杂度。这种做法不仅降低了模型对存储空间的需求，还有助于加速模型的训练和推理速度。然而，这种方法也存在潜在的风险，若剪枝过度，可能会损害模型的性能。此外，确定哪些神经元或连接是“不重要的”往往是一项挑战。知识蒸馏技术则提供了另一种视角，允许将大型模型中的知识有效地转移到更小型的模型中。这种方法在减少模型规模的同时，努力保持模型的性能，但它要求首先有一个大型且训练良好的模型，这本身就是一项成本较高的任务。同时，知识蒸馏的过程可能涉及复杂的训练策略，对于非专家来说可能较难掌握。量化技术通过降低模型参数的数值精度，如将 32 位浮点数转换为 16 位或 8 位，来减少模型的大小。这种方法能显著减少模型的存储需求并有可能加快推理速度，尤其是在支持低精度计算的硬件上。然而，量化可能会导致模型的精度降低，特别是在极低位数的情况下，因此需要仔细权衡精度损失和效率提升。除了模型本身的优化，硬件加速技术的发展也在提升大型神经网络的运行效率。使用 GPU、TPU 或专用神经网络处理器可以极大地加速模型的训练和推理过程。这些技术使得即使在硬件资源有限的环境中，也能有效运行复杂的深度学习模型。不过，这些高性能计算资源往往成本较高，可能不是所有研究者和开发者都能轻易承担。

综上所述，为了减少流量识别过程中因为硬件设施有限带来的困难，降低模型存储依赖，减少模型复杂度，节省模型训练时间，本论文重点展开针对轻量化的流量识别研究。

1.2　研究内容与创新性

根据上一节对流量识别存在的硬件资源有限的问题以及现有的研究方法的阐述和总结，接下来我们展示本论文使用的方法以及创新点。

本论文的研究重点在于探索在硬件资源有限的条件下，如何通过数据集优

化和模型轻量化来提升深度学习模型的效率和性能。随着深度学习技术的迅速发展，其在各个领域的应用逐渐普及，但这同时带来了对硬件资源的巨大需求。在许多实际应用场景中，尤其是边缘计算和移动设备领域，硬件资源的限制成为了一个不容忽视的挑战。因此，本论文旨在通过优化数据集处理和模型结构，降低深度学习模型对硬件资源的依赖，同时保持甚至提升模型的性能。

数据集优化：数据集优化是指通过各种技术和方法改善数据集的存储和处理效率，旨在减少存储和处理大量数据集时的压力。这个概念在机器学习和数据科学领域尤为重要，因为高效的数据集优化可以显著提高计算资源的使用效率，降低成本，并提高模型训练和数据分析的速度。数据集优化的几个关键方面包括：数据压缩，即通过算法减少数据所占用的存储空间。例如，使用文件压缩技术（如 ZIP）或者在数据级别应用更高效的编码和格式化技术；数据清洗，即去除重复、错误或不相关的数据，以减少数据集的大小，并提高其质量；数据降维，即使用技术如主成分分析（PCA）或自动编码器减少数据的特征数量，从而降低存储需求和提高处理速度；数据分片，即将大型数据集分割成更小的片段，以便在内存或存储资源有限的情况下更高效地进行处理；增量学习，即在数据流中逐步更新模型，而不是一次性加载整个大型数据集，从而减少存储和内存需求；采样，即从大型数据集中选择代表性的子集进行处理和分析，以减少处理的数据量（详细内容见第二章）。在数据集优化方面，本研究首先着重于特征选择的重要性。传统的深度学习模型通常涉及大量的数据处理，这不仅对存储资源构成压力，而且在训练过程中也会导致不必要的计算负担。为了应对这一挑战，我们提出了基于可解释性的精细化采集方法。这种方法不是简单地选择所有可用的特征，而是通过分析和理解数据，重点关注那些对模型性能有显著影响的关键特征。通过这种方式，可以显著减少数据的维度和大小，降低存储需求，同时保持甚至提升模型的准确性和效率。此外，精细化采集还有助于减少数据中的噪声和冗余信息，从而提高模型的泛化能力。

轻量化模型：模型轻量化（Model Lightweighting）是深度学习和机器学习领域的一个重要概念，旨在减少神经网络模型的大小，使其更适合在计算能力、存储空间或能源受限的设备上运行。模型轻量化的关键在于在尽可能少牺牲模型性能的前提下，减少模型的复杂度和计算需求。模型轻量化的几种常见技术包括：模型剪枝（Model Pruning）、知识蒸馏（Knowledge Distillation）、参数量化

（Parameter Quantization）、低秩近似（Low-Rank Approximation）、轻量化网络架构设计（Lightweight Network Architecture Design）（详细内容见第二章）。模型轻量化对于在移动设备、物联网设备以及边缘计算场景中部署深度学习模型至关重要。通过轻量化，可以使模型在资源受限的设备上运行，同时保持合理的性能。这不仅减少了计算资源的需求，还有助于降低能源消耗，是实现可持续 AI 的关键步骤。在模型轻量化方面，本研究探讨了减少深度学习模型中卷积层层数的方法。深度学习模型，尤其是卷积神经网络，在处理图像和视频等数据时表现出色，但这些模型往往包含大量的参数和层次，对计算资源的需求非常高。本研究通过实验发现，通过减少模型中的卷积层层数，可以在不显著牺牲性能的情况下，显著减少模型的大小和计算需求。这一发现对于在资源受限的环境中部署深度学习模型具有重要意义。通过轻量化的模型，我们可以在保持相对较高准确率的同时，使模型更适合在移动设备和边缘计算设备上运行。轻量化模型不仅降低了对计算资源的需求，还可以加快模型的训练和推理速度，这对于实时处理和响应至关重要。

研究目标：本论文的研究目标是探索在硬件资源受限的条件下，如何有效地通过数据集优化和模型轻量化来提升深度学习模型在流量识别方面的效率和性能。随着深度学习技术的快速发展和广泛应用，对硬件资源的需求日益增长，尤其在边缘计算和移动设备等资源有限的环境中。本研究的目的是通过改进数据集处理方法和简化模型结构，减轻深度学习模型对硬件资源的依赖，同时保持或提升模型性能。

研究方法：针对以上的研究目标，我们结合上一小节提炼出的相关问题，列出了对应的解决方法和创新点。

第一，针对深度学习流量识别研究的**数据集优化**方法，我们在搭建深度神经网络后，我们测试了多种模型解释性方法，特别是采用了 **SHAP（SHapley Additive exPlanations）**算法在随机森林中绘制决策图，作为深度模型的可解释性工具。这种方法细化了模型解释的粒度，使我们能够更透明地理解模型如何进行分类决策，并计算出模型对不同特征的依赖程度。通过这种方式，我们不仅提高了模型的可解释性，而且还通过精细化采集那些对模型贡献度高的特征，有效地减少了数据集的存储占比，从而优化了整个数据集。

第二，针对深度学习流量识别研究的**模型轻量化**，我们将具有相同五元组

（源 IP 地址、目的 IP 地址、源端口、目的端口和协议类型）的数据聚合为一个矩阵样本，利用深度学习网络（如 ResNet）处理图像的能力，将这些矩阵样本视为图片进行训练。ResNet 通过其特有的跳跃连接（Residual Connections）解决了梯度消失/爆炸问题，使得我们能够训练更深层次的网络。这种结构不仅保证了模型在流量识别任务上的准确性，还提高了其泛化能力、灵活性和鲁棒性。最后，为了实现模型轻量化的目标，我们通过采用**知识蒸馏 (Knowledge Distillation)** 的方法，设计了一个更少数量的卷积层构建 student 神经网络，减少了每层卷积的计算时间和模型的整体训练时间。这样的轻量化模型不仅减轻了对计算资源的需求，还保持了足够的性能，以满足实时流量识别的需求。

1.3 研究的意义

理论意义：本论文的研究重点在于探索如何在硬件资源有限的条件下通过数据集优化和模型轻量化提升深度学习模型的效率和性能。在数据集优化方面，我们采用了基于 SHAP 的决策图和精细化特征采集的方法，这不仅提高了模型的可解释性，还通过关注对模型性能有显著影响的关键特征，有效地减少了数据集的存储占比。在模型轻量化方面，研究聚焦于将流量数据视为图像样本并利用 ResNet 网络进行处理，这种方法通过减少卷积层的数量，在保持模型性能的同时减少了模型的大小和计算需求。

现实意义：深化了对数据集优化和模型轻量化的理解，还扩展了深度学习在资源受限环境下的应用范围。在实际应用中，这些研究成果提高了深度学习模型在资源有限环境中的可行性，增强了模型的可解释性和透明度，并提升了流量识别技术的性能和效率。这对于网络安全和网络管理等领域具有深远的影响，特别是在边缘计算和移动设备等硬件资源受限的环境中。

1.4 论文内容组织结构

如图 1.1 所示为本学位论文的结构图，第一章介绍了本论文的研究背景、问题、研究内容和创新点。第二章介绍了现有的研究方法，同时阐述了对应研究方法的不足，进而总结出对应的研究问题。第三章介绍了应用数据集的采集和处理方法，展示了数据集样本数据包的内部架构和数据样本的处理流程，值得注意的是为了实验方案的全面，本章节阐述了分别从应用层和传输层采集数据，使用处

理过后的加密流量与未加密流量进行实验。第四章介绍了基于深度学习的流量识别方法，展示了如何筛选合适的深度神经网络并且结合第三章处理出的数据实现流量识别任务；与现有的方法相比，本章节在确保识别准确率的同时，探索了模型分类对数据报头字段和数据字段的关注程度。第五章介绍了基于可解释性算法的数据轻量化方法，展示了**研究点**一的具体研究方法和评估结果；与现有的方法相比，本章节描述的方法减少了额外的处理处理开销，保证了对重要特征的提取，结合可解释性算法筛选重要特征，从而减少了数据集的大小。第六章介绍了基于知识蒸馏的模型轻量化方法，展示了**研究点**二的具体研究方法和评估结果；与现有的方法相比，本章节描述的方法在研究点一数据轻量化的基础之上，结合模型轻量化方法简化了模型的复杂程度。第七章展示了我们对论文的总结和对未来工作的展望。

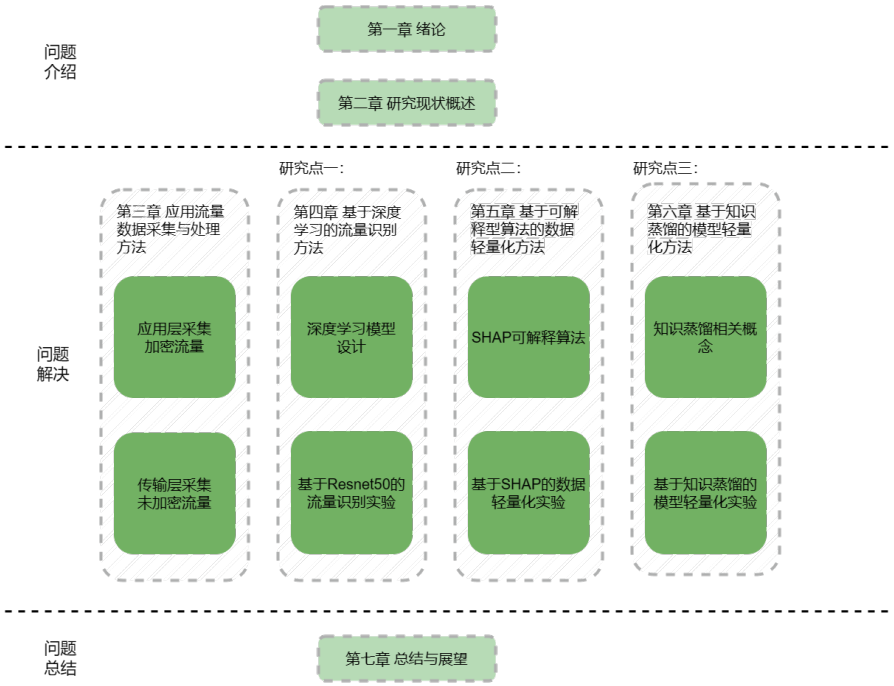

图 1.1 论文结构图

第一章，阐述了流量识别的研究背景、现有的问题、本论文的研究内容和创新点、以及研究的意义。

第二章，展示了论文所需要的基础知识；分析了流量识别方法和模型轻量化的发展现状，对现有的方法做了详细的介绍和分析，阐述了它们的优点和不足之处；最后凝练出本论文需要研究的科学问题。

第三章，展示了本论文的研究对象神经网络的输入数据集；详细阐述了应

用层和传输层流量数据采集与梳理的方法，描述了数据 pcap 包处理的程序框架，减少了复杂字段的识别时间，提高了数据包处理的速度。目的是提高模型执行效率以及便于在模型解释时确定物理含义。

第四章，详细阐述了基于深度学习的流量识别方法，分析了使用深度学习进行流量识别的研究现状，描述了识别所需要的深度学习框架，并将识别结果采用交叉熵验证的方式进行了测试，目的是验证确保识别结果的可靠性。本章节在使用模型进行流量识别的基础上，探索了流量报头字段和数据字段对识别结果的重要性，方便后续的特征筛选。

第五章，详细阐述了基于可解释性算法的数据轻量化方法，分析了可解释性算法的相关概念，重点介绍了 SHAP 可解释性算法。本方法提升了各个特征的对识别结果贡献度的透明度。本章节描述了 SHAP 可解释性算法的算法框架，并将筛选出的特征序列采用交叉验证的方式进行了验证，验证了所提出的方法对数据轻量化的可靠性。

第六章，详细阐述了基于知识蒸馏的模型轻量化方法，描述了模型轻量化以及知识蒸馏的相关概念。方法描述了知识蒸馏中教师-学生模型的相关框架，该方法在确保流量识别准确率的同时，减少了模型卷积层的数量，降低了模型复杂度，减少了模型参数，从而提高了模型的泛化能力，减少了模型对外部环境的依赖。

第七章，对全文进行了总结与展望，总结了本论文的主要的创新点和贡献，并提出了未来的研究方向和挑战。

第 2 章　研究现状概述

本章节首先展示了流量识别的任务定义，从任务的目标角度对常见的流量识别任务进行了介绍，并对流量识别常用方法进行了总结。其次，展示了针对模型轻量化不同的处理方法。最后我们介绍了基于深度学习模型进行识别的相关概念，并总结了现有工作的优点和不足之处。最后，提炼出了需要解决的问题。

2.1　流量识别技术研究

流量识别旨在从网络流量中分类出特定类别的流量，可以是特定应用的流量，应用的行为，恶意流量等等。流量识别作为一个传统的网络技术，已经研究了二十多年，并在入侵检测系统等安全应用中广泛使用。传统的流量识别根据端口号，协议等特征进行识别，但近年来，出于对安全和隐私的考虑，加密流量激增，为流量识别带来了新挑战。在本文中，我们对现有的流量分类技术进行了介绍，不仅包括传统的技术，还包括最近的技术和趋势。根据流量意图的不同，流量在传输过程中会带有很多特征，一些工作利用单个数据包的特征对流量进行识别，一些工作则对整个网络数据包流进行处理，这会根据任务具体的目标不同而有所变化。进行流量识别的第一步是明确识别的目标，对于不同的任务，由于数据的特点和流量的意图不同，所采用的方法也有所不同。

流量识别旨在识别流量的意图与类别。在这里流量指的是在网络中传输的数据包。其源头是应用层的读写操作 (结构如图 2.1 所示)，经过传输层协议的变换（分片、协议状态机、加密等），流量序列产生一定变化。但是这种变化非常有限，因为流量的发生过程本质是确定性的，随机因素较小，因此对于特定环境中的特定应用（浏览器访问 google.com）各种流量特征体现出相当大的一致性和独特性，这就使“从流量特征识别应用”的监督学习问题成为可能。流量识别在网络安全和管理中扮演着关键角色，涉及多种任务，如应用流量分类、网站指纹识别、应用行为分类、混淆流量识别及恶意流量识别。应用流量分类通过区分特定应用的流量，助力于流量审查与网络秩序维护，但也可能影响用户隐私。网站指纹识别旨在根据流量识别用户访问的网页，而应用行为分类进一步分析用户行为。混淆流量识别关注于识别经过伪装的流量，以保护隐私或安全。恶意流量识

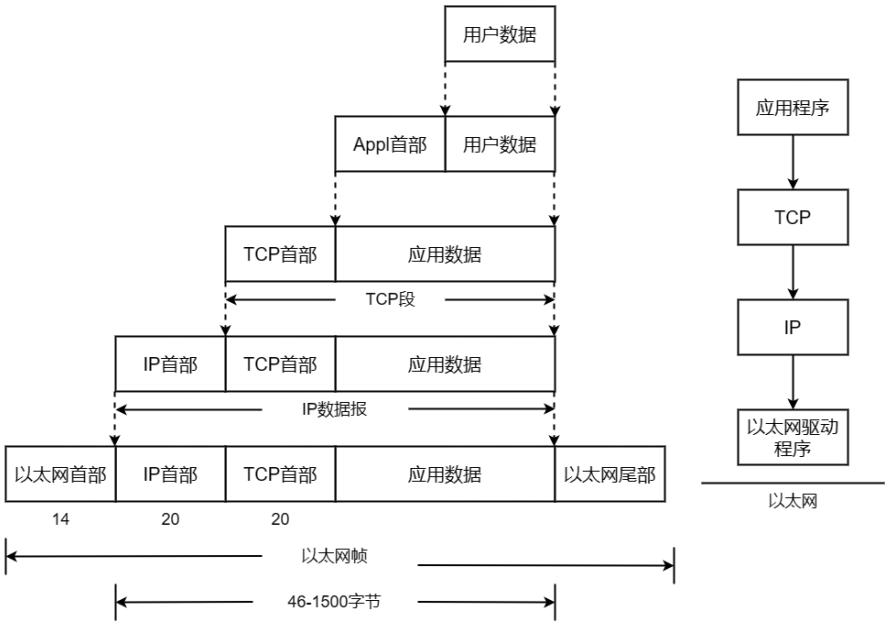

图 2.1 流量结构图

别则致力于识别和防止网络攻击。这些任务通常结合机器学习和深度学习技术，如 RNN、LSTM、CNN 等，应对加密流量带来的挑战。尽管实验室条件下识别准确率高，但在真实环境中可能面临数据分布不一致的问题。这些研究表明，即使数据包加密，用户的隐私和流量意图仍可能被识别。

目前常用的流量识别方法就是采用特征工程的方式，对原始数据进行一定的预处理和清晰，根据任务目标抽取各种特征，采用专门设计的算法进行识别。过去比较常用的方式是采用字段匹配和正则表达式，在加密流量越来越普遍的现在，越来越多的工作利用统计特征，采用机器学习和深度学习的方式训练分类器进行识别。还有一些工作进行了无监督的方式，例如-means、DBSCAN 或层次聚类算法，这些方法通过分析流量特征将数据分组，不需要预先标注的数据。主成分分析（PCA）可用于识别流量数据中的主要成分和模式。自编码器，通过重构输入数据来学习数据的有效表示。高斯混合模型（GMM），用于建模具有多个子分布的流量数据。下面将对流量识别中的常用方法和特征进行介绍。

传统报文检测会分析 IP 包的源地址、目的地址、源端口、目的端口以及协议类型。这种方式可以检查端口是否被正常使用，比如 443 端口是否传输的是明文流量。DPI（Deep Packet Inspection）深度包检测技术是在传统报文检测之上增加了对应用层数据的协议识别，通过解析应用层数据来确定流量对应的业务信息。传统报文检测和深度包检测都可以对流量进行一定程度的识别，但很多时

候，对于加密流量和混淆后的流量难以检测。有很多工作都是这种基于规则的思想，比如基于证书的检测，基于协议首部字段的检测，基于通讯模式的检测，对于明文数据是一种极为有效的手段，但人工分析特征需要专家得知识，并且很容易被拼接，改造后的流量规避，对加密流量这样的手段起到的作用也是有限的。近几年有不少工作都采用了这种类似规则匹配的方式，[Shbair2016] 通过对比 SNI 和 IP 对应的域名信息来增强防火墙的安全性。[Husák2016] 建立了 SSL / TLS ciphersuite list 和 HTTP User-Agent 的字典，并将 User-Agent 分配给观察到的 SSL / TLS 连接，用来识别通信的客户端。[Papadogiannaki2018]，从流量中提取行为特征，如包出现频率或包所在的位置，采用正则匹配固定模式，可以对 facebook 中的消息，语音和视频流量进行区别。

统计特征通常是对于整条数据流进行处理 [Rezaei2018]，例如流长度，平均包长，流开始和结束时间等等，也可以跟多个包之间的关系比如最小包间时延，包上下关联属性等。用统计特征不仅可以对明文传输的流量进行处理，还可以针对加密流量，基于统计特征的机器学习分类器已经成为识别加密流量的主要手段，但统计特征并非是完美的，比如混淆流量和恶意流量可以通过字段填充或伪装改变一定的统计特征。并且很多工作用到的统计特征需要对整条流进行处理，也就意味着很难做到实时的检测，如果仅能用于离线分类，应用场景会很有限。除此之外，统计特征还需要人工设计和构建，依赖专业的知识和经验，这一点明文特征也要相同的缺陷，特征设计的好坏会影响最终识别的效果，如何构造的统计特征是这个问题的难点。很多工作都是采用的统计特征，像 [K. Li2018] 采用 HTTP 请求的统计特征检测来自恶意软件的安全威胁，[Anderson2017] 也是采用机器学习对统计特征训练了分类器，他们同时还考虑了带噪声的标签。分类器的选择也会对识别的结果有着很大影响，比较常用的方法有决策树，随机森林，朴素贝叶斯，Adaboost，深度森林等等，这些算法有着各自的特点和优势，在 [Namdev2015] 这篇综述中，对机器学习的方法在流量识别中的应用进行了介绍。流量进行加密处理后，调查基于规则匹配的方法已经逐渐失效，而许多工作中采用机器学习的方法训练分类器，需要人工提取特征，而这需要相关领域的专业知识，并且不同协议和应用构造的特征可能是不同的，在此基础上，一些工作将 nlp 中发展迅速的深度学习方法应用在了流量分类上，采用表示学习的思想可以避免繁琐的规则构造，从加密的原始信息中自动提取关键信息并生成有区分

性的加密流量指纹。

对于采用深度学习方式进行流量识别的过程，[Rezaei2018] 的综述中已经进行了较为详细的论述，简而言之，就是在这方面也有很多经典的工作，[Lotfollahi2017]: 将 payload 从 tcp/udp 层开始对其填充，保证协议头部分 20 字节和 8 字节对齐，对于负载部分，采用前 1480 个字节，对于不足的负载进行 0 填充，保证维度一致性，使用 ANN 和 SAE 进行特征提取生成指纹。[Rimmer2018]: 利用匿名化网络 tor 的特性，采用长度序列的方向序列作为深度学习网络 SAE、CNN、LSTM 的输入，从而分类访问的网页。[Liu2019] 提出了一种基于表示学习的流序列网络，以双向 GRU 为基本单元，编码解码为整体结构，同时采用重构机制增强加密流量指纹的表现能力，自动化的从原始流量信息中学习有效特征，提升了分类的精度。尽管基于深度学习的识别方法在工作上显示出了很好的效果，但是这种方式往往是有监督的方式，需要大量的标注数据，对于数据的搜集而言是一个难点。

在实验室环境中，为了从强调方法的效果，往往是采用单一的算法进行识别，在现实的应用场景中，工业界中网络流量分析（NTA）大多结合使用机器学习、高级分析和基于规则来检测企业网络上的可疑活动。马尔可夫概率指纹（Mampf）是一种高效的加密流量分类方法，通过利用马尔可夫链模型捕捉流量数据中数据包之间的转移概率。这种方法特别适合处理加密流量，因为它不依赖于数据包的具体内容，而是专注于流量模式。然而，它可能在处理高度随机或非常规模式的流量时效果不佳。SFC（Stateful Flow Clustering）方法通过分析流量的时间和空间特征进行聚类，优点是能够识别出具有相似行为模式的流量，从而提高识别精度。但这种方法可能需要较多的计算资源，并且对于流量特征的选择敏感。n-gram 熵方法通过计算流量数据的熵来评估其复杂性和随机性，这对于理解和分类流量非常有用。它的优点在于对数据内容的要求较低，但可能无法准确区分具有相似熵值但不同行为的流量。

总的来说，这些技术各有优缺点，选择合适的技术取决于特定应用场景和数据特性。虽然这些技术在实验环境中的性能表现逐渐提升，但在外部资源受限的情况下（例如内存、显存、缺少 GPU 等），如何在保持准确性的同时降低对外部环境的依赖，成为另一个需要考虑的问题。

2.2　模型轻量化相关概念

深度学习模型轻量化方法旨在通过减少模型参数、降低计算复杂度和减小存储需求等手段，在保持较高性能的同时实现模型的轻量化。这些方法可以分为几类，包括剪枝、低秩分解、量化、知识蒸馏、紧凑网络架构、稀疏性和其他一些相关方法。它们之间的关系和差异主要体现在降低模型复杂度的策略、压缩程度和性能损失等方面。**剪枝**方法通过移除不重要的神经元或连接来减少模型参数；低秩分解则利用矩阵分解来降低模型参数数量；**量化**方法将权重和激活值用较少的比特数表示以减小存储和计算需求；**知识蒸馏**将一个大型教师模型的知识迁移到一个较小的学生模型；**紧凑网络架构**通过设计更高效的网络结构来降低模型复杂度；**稀疏性方法**则尝试在模型中引入稀疏性，以减少参数数量和计算量。此外，还有一些其他的轻量化方法，如**神经结构搜索（NAS）**、**提前退出（early exiting）**，它们在不同程度上结合了上述方法的优点。在接下来的部分，将详细介绍这些轻量化方法的原理、优缺点和应用场景。

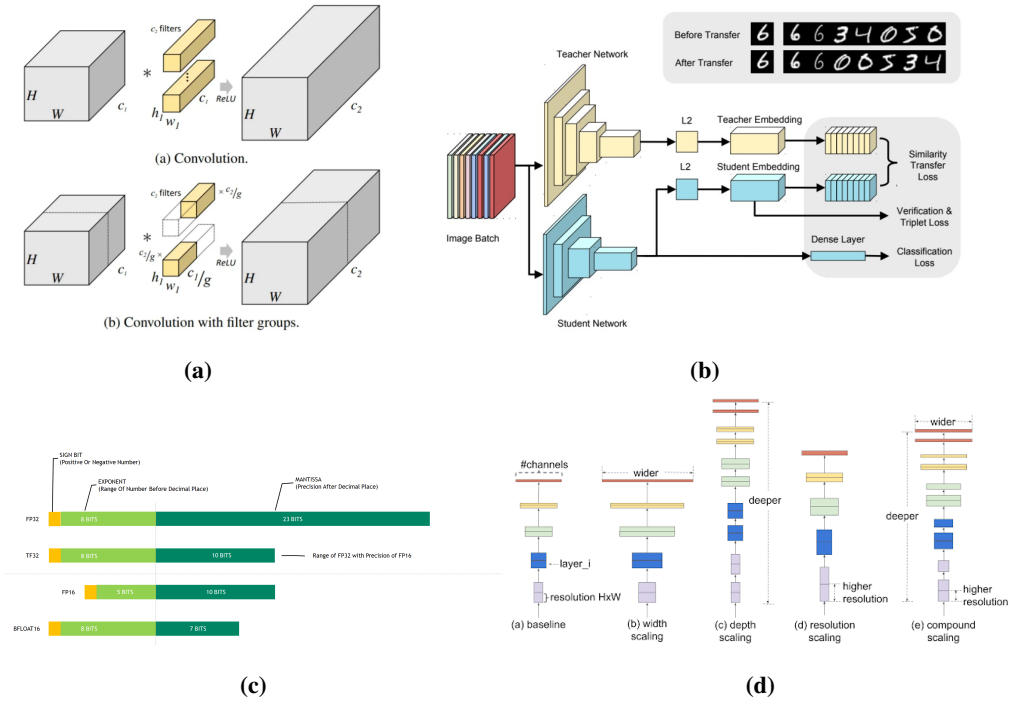

图 2.2 (a) 分组卷积剪枝 (b) 知识蒸馏 (c) Tensor Float 32 (d) 紧凑网络架构

Figure 2.2 (a) Pruning (b) Knowledge distillation (c) Tensor Float 32 (d) Compact network architecture

剪枝作为深度学习模型轻量化的一种常见方法，旨在通过减少模型参数、降

低计算复杂度和存储需求来实现模型的精简化。主要分为结构化剪枝和非结构化剪枝两类。结构化剪枝针对整个神经元、通道或层进行裁剪，以保留模型的稀疏结构，便于硬件加速（如图 2.2(a) 所示）。通道剪枝通过删除整个通道来降低参数数量，层剪枝则涉及删除整个网络层，而分组卷积剪枝则通过将卷积层的输入和输出通道分组，以降低计算复杂度。非结构化剪枝则基于权重重要性，通过删除单个权重来压缩模型，具有更高的压缩率，但可能导致不规则的稀疏性，难以直接利用硬件加速。在非结构化剪枝中，常采用设定阈值并剪枝小于该阈值的权重的策略，以降低模型参数数量。此外，条带剪枝是介于结构化剪枝和非结构化剪枝之间的一种方法。它通过剪枝连续的一组权重形成“条带”，在保持一定规整性的同时提供更细粒度的剪枝控制，有助于减少对模型性能的影响并提高剪枝后模型的稀疏性。综合而言，剪枝方法在轻量化深度学习模型方面发挥着重要作用，通过有效减少参数数量和计算复杂度，为在资源有限的环境中部署模型提供了可行的解决方案。

量化技术是指将连续的数值或信号转换为离散的数值或信号的过程（如图 2.2(c) 所示）。在机器学习和深度学习领域中，量化通常用于减少神经网络中参数的位数和精度，以降低模型的存储和计算开销。通过将参数量化为较低位数的整数或浮点数，可以大幅减小模型的大小并提高模型在移动设备等资源有限的环境下的性能。同时，量化技术也可以用于降低神经网络计算过程中的功耗和延迟，从而提高模型的运行速度。量化技术是深度学习中的一个重要技术，在很多实际应用中具有很高的实用价值。在神经网络中，量化指的是将模型中的浮点数参数转换为更小的整数或固定位宽的数字，以便于存储和计算。这通常可以通过减少模型中参数的位数来实现，从而降低内存和计算开销，并提高推理速度。量化技术可能会对模型的精度造成一定的影响，因为它减少了参数的精度。在实际应用中，模型的参数通常是 FP32 单精度，我们可以使用整数、FP16、TF32、MPT 等代替。

知识蒸馏是一种深度学习模型轻量化方法，旨在通过将一个大型教师模型的知识迁移到一个较小的学生模型，从而实现在保持较高性能的同时减小模型大小和计算复杂度。知识蒸馏的基本原理是让学生模型学习教师模型的输出分布，从而吸收教师模型的知识（如图 2.2(b) 所示）。这个过程主要包括几个关键步骤。首先，需要准备适用于任务的训练数据集，该数据集可能与教师模型的训

练数据集相似。接着，选择一个强大的大模型作为教师模型，并确保在任务上达到了满意的性能。然后，定义一个较小的学生模型，通常拥有比教师模型更少的参数。设计知识蒸馏损失函数时，要考虑任务的交叉熵损失以及教师模型输出与学生模型输出之间的距离，常见的损失函数包括温度缩放的交叉熵和 KL 散度项。在训练过程中，使用知识蒸馏损失来指导学生模型的训练，通过迭代调整学生模型的权重。最后，通过在独立的验证集上评估学生模型的性能，调整超参数，保存训练好的学生模型，以备将来使用。这个过程使得学生模型能够借鉴教师模型的知识，实现在相对较小的模型体积下取得更好性能的目标。

紧凑网络架构作为一种深度学习模型轻量化方法，注重通过设计更高效的网络结构来在减小模型复杂度的同时保持较高性能（如图 2.2(d) 所示）。在这一领域，EfficientNet 是一种备受关注的卷积神经网络（CNN）模型，由谷歌研究团队于 2019 年提出。EfficientNet 的设计原理基于一种称为 Compound Scaling（复合缩放）的优化策略，该策略在网络深度、网络宽度和输入分辨率三个方面进行了综合调整。首先，复合缩放中的网络深度体现在对网络层数的调整，通过增加深度，模型可以更好地捕捉抽象层次的特征，提高性能。然而，研究者们也认识到过深的网络可能引发梯度消失、爆炸以及过拟合等问题。其次，网络宽度的调整涉及每层网络中通道数或神经元数量的变化。增加网络宽度有助于提升模型容量，但需注意避免过宽的网络导致计算资源的浪费。最后，输入分辨率的调整关注图像输入尺寸的变化，通过增加分辨率，模型可以更好地捕捉细节信息，然而，过高的分辨率会增加计算复杂度和内存消耗。通过在这三个维度上进行平衡的复合缩放，EfficientNet 取得了出色的性能，并且在各种图像分类任务中表现卓越，同时具备较高的参数和计算效率。这种综合调整的设计策略使得 EfficientNet 成为紧凑网络架构领域的翘楚，为深度学习模型在资源受限环境中的应用提供了有力的解决方案。

稀疏性在深度学习模型中的应用是为了减少模型的参数数量，从而实现模型的轻量化。这一概念可以从两个角度来理解，即原始域稀疏性和变换域稀疏性。首先，原始域稀疏性指的是在模型参数的原始表示中存在大量的零值或接近零值的元素。这种稀疏性可以通过一系列方法实现，如权重剪枝和正则化等。在训练过程中，我们可以使用稀疏矩阵来表示权重，仅存储非零权重值，从而显著减少模型的存储需求。同时，引入 L1 正则化等稀疏性惩罚项，有助于鼓励模

型学习稀疏的权重，逐步消除不重要的连接，降低计算复杂性。为了提高计算效率，还可以利用专门针对稀疏矩阵操作的算法和库。其次，变换域稀疏性是将模型参数从原始域转换到另一个域，以便更容易地识别稀疏性。这包括将权重矩阵转换为频率域，如傅里叶变换或小波变换。在频率域中，通过阈值方法或其他技术去除不重要的参数，再将剩余参数转换回原始域，实现模型参数的压缩。小波变换的优势在于可以适应非稳态信号变化的频率需求。总体而言，稀疏性的引入在模型设计中起到了优化存储和计算的作用，为轻量化模型提供了有力支持。通过在训练过程中优化模型的参数表示，我们能够更加高效地利用有限的计算资源，从而在实际应用中取得更好的性能。

神经结构搜索（Neural Architecture Search，NAS）代表着一项自动化技术，其目标是简化神经网络结构的搜索和优化过程。传统上，神经网络的设计是一项复杂且耗时的任务，通常需要领域专家进行手动设计和调整。然而，NAS 通过自动搜索的方式使得机器学习系统能够自主发现最优网络结构以解决特定问题。NAS 的工作原理是在预定义的搜索空间内进行探索，试验不同的网络结构和超参数配置。该搜索空间定义了可用的网络层、连接方式以及其他潜在的配置。接着，NAS 采用优化算法（如强化学习、进化算法、贝叶斯优化等）来探索这些潜在组合，并根据它们在训练数据上的性能进行评估。在搜索过程中，优化算法根据模型在数据上的表现来调整搜索方向，逐步发现更优的网络结构。经过足够的搜索和评估时间，NAS 最终能够发现一种网络结构，该结构在给定任务上表现出色。这种自动化方法节省了大量时间和资源，并有时能够找到超越人类专家设计的神经网络的解决方案。然而，NAS 也存在一些局限性。其搜索过程可能需要大量计算资源和时间，特别是在庞大的搜索空间中。此外，NAS 找到的网络结构可能难以解释和理解，这可能对网络的可解释性和可维护性产生影响。尽管如此，NAS 依然是一个备受关注的研究领域，有望推动神经网络设计的自动化和优化。

提前退出（Early exiting）作为一种深度学习模型中的技巧，尤其在神经网络中广泛应用。其核心目标在于在不牺牲模型性能的前提下，降低计算复杂度，提高模型计算效率。在深度神经网络中，通常包含多层神经元，传统的前向传播过程需要经过所有层的计算，最终生成预测结果。然而，有时模型在经过部分层的计算后已经能够产生足够准确的预测。在这种情况下，继续进行更深层次的计算

可能会导致计算资源的浪费。为解决这一问题，提前退出策略被引入，其中某些中间层被设定为潜在的输出层。这意味着，当输入数据通过这些中间层时，可以根据一定标准（例如置信度或损失函数阈值）判断预测结果是否已足够可靠。一旦满足退出条件，模型即可提前终止计算并输出当前层的预测结果。这种灵活性使得模型能够在不牺牲预测质量的前提下，避免进行多余的计算，从而提高整体效率。综合而言，提前退出作为一项深度学习技巧，为神经网络的计算效率提供了有效的解决方案。通过在适当时刻终止计算流程，模型能够更加智能地利用计算资源，使其在实际应用中更为高效。

轻量化深度学习模型的方法包括剪枝、低秩分解、量化、知识蒸馏、紧凑网络架构和稀疏性，每种方法都有其独特的优势和限制。剪枝通过删除冗余参数降低模型大小，提高计算效率，但需要额外的训练迭代，非结构化剪枝可能导致不规则稀疏性。低秩分解通过分解权重矩阵减少参数数量，节省存储空间，但可能引入信息损失，对超参数敏感。量化显著减小模型大小，提高在资源受限环境下的性能，但可能对模型精度造成影响，需要额外的训练过程。知识蒸馏在减小模型大小的同时保持相对高性能，但需要选择适用的大模型作为教师，损失了一些可解释性。紧凑网络架构通过设计高效结构在减小模型复杂度的同时保持较高性能，适用于资源受限环境，但在某些任务上可能不够灵活。稀疏性通过减少模型参数数量实现轻量化，但原始域稀疏性可能影响模型精度，变换域稀疏性需要复杂的转换过程。选择合适的轻量化技术应基于具体应用场景、资源限制和对模型性能的要求，可能需要权衡不同技术之间的优缺点，本研究着力于选择合适技术实现模型轻量化。

2.3　本章小结

本章节我们介绍了流量识别和模型轻量化的相关基础知识。然后我们分析了现有的流量识别方法和模型轻量化方法，并对现有的方法进行了比较和讨论。对于流量识别的方法，我们分析了传统的方法存在准确率不高等问题。对于模型轻量化的问题，我们分析了现有的模型轻量化方法中剪枝、低秩分解、量化、知识蒸馏、紧凑网络架构、稀疏性等技术的原理以及优缺点，然后根据实验环境选择合适的技术。

第 3 章　应用流量数据采集与处理方法

3.1　数据集介绍

在网络流量数据集方面，很难找到一个适用于多个领域的优秀数据集，但仍有一些工作将自己的数据集进行公开并且被广泛的使用，加拿大网络安全研究所 CIC 发布了几个经典数据集，数据集可以用 CIC 提供的 CICFlowMeter-V3 对特征进行提取。[Gerard2016] 的 VPN-nonVPN 数据集是由加拿大网络安全研究所 (ISCX) 提供的，涵盖了 VPN 和 nonVPN 流量，具有标签的 7 大类网络服务。这个数据集的优点在于提供了有关加密虚拟私人网络（VPN）和普通网络（nonVPN）流量的信息，使得研究者能够对这两种流量进行深入分析。然而，缺点可能在于数据集的规模相对较小，可能不足以覆盖所有网络行为的变化和复杂性。[Iman2018] 的 CSE-CIC-IDS2018 on AWS 数据集是由通信安全机构 (CSE) 和加拿大网络安全研究所 (CIC) 合作提供的，包括七个不同的攻击场景，涵盖了各种网络攻击类型。这个数据集的优点在于提供了多样性的攻击场景，有助于研究网络入侵检测系统的性能。然而，缺点可能在于数据集的规模和复杂性，可能需要更多的计算资源和专业知识来处理和分析。[Laya] 的安卓恶意流量数据集包含 5000 多个软件的流量，其中有 426 个恶意软件，并将这些恶意软件的流量分为了 4 类。这个数据集的优点在于专注于安卓平台的恶意流量，有助于研究和分析安卓恶意软件的行为。然而，缺点可能在于数据集规模相对较小，可能无法完全涵盖所有安卓恶意软件的变种。

这些数据集各有优缺点，研究者在选择使用时需要根据研究目的和需求权衡各方面的考虑。目前的很多工作都是在实验室的环境中测试，尤其是恶意流量识别和网站指纹等任务在真实环境中准确率和召回率都会下降很多。并且流量识别技术在落地部署时，还需要考虑实时性，这也就意味着基于数据流的方法都会失效，仅仅用几个数据包进行识别也会导致较高的假阳性。很多工作都是采用自己收集的数据集，比如应用识别和行为识别往往都是针对特定的应用或者特定领域使用，所以没有公开的数据集资料，还有一些工作是利用一些软件生成的数据，由于收集软件的要求，不适合公开。由于真实网络环境下的流量是变化的，一些数据集比如 KDDCUP99 和 NSL-KDD[Tavallaee2009] 等已经和当前的真

实网络环境差异较大。综上所述，本实验选择采用自己的数据集继续进行训练。

为了进行基于深度学习的流量识别技术研究，可以通过 PCAP 包捕获、NetLog、Packetbeat、Wireshark 等方法来获取网络流量数据。NetLog 工具是一种 Android 监控工具，能够监控来自用户和系统应用的所有网络访问尝试和通信。NetLog 可以为每次网络访问生成日志，这对于精确地标记移动网络流量的追踪数据非常有用。通过这种工具，可以方便地构建用于应用程序识别（APP-ID）的真实网络移动流量数据集。WiKibookBibliography。Packetbeat 是一个轻量级的网络包分析器，它可以从主机和容器发送数据到 Logstash 或 Elasticsearch。Packetbeat 支持多种应用层协议，并且可以将网络流量数据实时地整理并发送到 Elasticsearch 或 Logstash 中，方便进行数据分析和监控。WiKibookBibliographyWireshark 是一款流行的网络包捕获和分析软件，它可以识别超过 2000 种协议。Wireshark 的命令行工具 tshark 可以用于实时捕获流量或读取和解析捕获文件。tshark 支持将解析后的包数据以 JSON 格式输出，适用于 Elasticsearch Bulk API，从而可以方便地将网络包数据导入到 Elasticsearch 中进行分析。WiKibookBibliography。

在考虑基于深度学习的网络流量识别技术所使用的数据获取方法时，我们可以看到每种方法都具有其独特的优势和局限性。首先，PCAP 作为网络流量捕获的标准格式，因其广泛使用和强大的兼容性而受到青睐，但同时也面临着数据量庞大和可能涉及隐私问题的挑战。NetLog 工具，尽管在精确标记移动网络流量方面表现出色，适合于移动设备的网络流量分析，但其主要用于 Android 平台，可能不适用于其他操作系统，并且背景流量的干扰可能导致数据噪声。在使用 Elastic Stack 中的 Packetbeat 时，其优点在于能实时处理网络流量数据并支持多种应用层协议，但它不适合全包捕获，且配置可能较为复杂。最后，结合 Wireshark 和 Elastic Stack 的方法虽然在协议识别能力上强大，适用于复杂网络环境，但可能产生的庞大数据量和高技术分析要求也是其不容忽视的缺点。因此，选择适合的方法应基于具体的研究需求和环境，综合考虑各种方法的优势和局限性。

3.2 PCAP 相关概念

PCAP（Packet Capture，数据包捕获）是网络分析和网络安全中的关键工具，它提供了详细的网络流量记录，适用于多种应用。PCAP 包括捕获和分析网络流

量数据，通常存储在 PCAP 或 pcapng 文件格式中。PCAP 文件包括一个文件头和多个数据包记录，每个记录代表从网络捕获的单个数据包。这些记录包含时间戳、数据包长度和网络上传输的实际数据包长度等重要细节WiKibookBibliography。

不同版本的 PCAP，如 Libpcap、WinPcap、PCAPng 和 Npcap，针对不同操作系统和用例，每种都提供了独特的网络监控和分析能力。像 Wireshark、Nmap 和 Snort 这样的工具使用这些格式来捕获和分析网络流量，其中 Wireshark 尤为突出，因为它能够使用 PCAP 文件来过滤流量和解决网络性能问题WiKibookBibliography。WiKibookBibliography。在取证调查中，PCAP 分析具有极高的价值，它使网络事件的重建成为可能，从而追踪安全事件的来源和影响。它在恶意软件分析和逆向工程中也发挥着重要作用，同样在优化网络性能方面也很关键，比如通过识别瓶颈和解决延迟问题WiKibookBibliography。PCAP 的起源可以追溯到 20 世纪 70 年代，当时开发了伯克利数据包过滤器（BPF），后来在 1990 年代开发了 Tcpdump 工具，为 PCAP 作为标准文件格式奠定了基础。在 20 世纪 90 年代末引入的 PCAP 文件格式，已成为捕获和分析网络流量数据的事实标准WiKibookBibliography。在现实世界中的应用方面，PCAP 有助于检测恶意软件感染、调查数据泄露和解决网络性能问题。它提供了网络流量的详细视图，使专业人员能够识别安全威胁、进行调查，并优化网络资源。在网络安全领域，PCAP 分析的专业知识具有很高的价值，专业人员通常担任网络分析师、安全分析师、事件响应者或数字取证调查员等职位WiKibookBibliography。尽管 PCAP 具有许多优势，但它也有限制，特别是在加密流量方面，这可能会绕过数据包嗅探器，以及在捕获整个网络通信方面，特别是如果嗅探器在网络中的位置不够策略性。加密和数据包嗅探器的战略性布局是有效利用 PCAP 进行网络安全和分析的关键因素WiKibookBibliography。

总之，PCAP 是网络分析中的基础技术，提供了对网络流量的深入洞察，用于安全性、故障排除和优化目的。多年来，它的发展和演变使其成为网络安全领域不可或缺的工具。

3.3　PCAP 数据包架构

PCAP（Packet Capture）数据包的内部结构按照网络协议分层结构组织，其中包含了链路层、网络层、传输层和应用层的关键信息。根据”PCAP file format”（来自 The Tcpdump Group 的官方网站）所述，首先是数据链路层头部（Data Link

Layer Header），这一部分包含了数据包在数据链路层的头部信息，其中可能包括了源和目标 MAC 地址、帧类型（如以太网、Wi-Fi 等）、帧起始和结束标志等信息。这一部分的结构会根据所捕获的网络类型和协议而有所不同。接着是网络层头部（Network Layer Header），通常是 IP 头部信息，包括源和目标 IP 地址、IP 版本、服务类型等。若存在传输层数据，其头部信息紧随网络层，如果在数据链路层捕获的数据包是 IP 数据包，接下来的部分就是网络层的头部信息。这可能包括了 IP 头部信息，如源和目标 IP 地址、IP 版本、服务类型、生存时间（TTL）等。如果数据包在网络层是 IPv6，那么头部结构也会相应有所不同。对于传输层头部（Transport Layer Header），包含了源端口、目标端口、序列号、确认号、标志位等 TCP 头部信息。对于 UDP 数据包，会包括源端口、目标端口、长度和校验和等信息。数据部分是数据包的有效载荷，可能是各种应用层数据，如 HTTP 请求或响应。此外，参考研究论文”Libpcap and WinPcap: Applications, Challenges, and Progress”（Mobicom 2006）指出，PCAP 数据包还可能包含附加信息，如时间戳、数据包长度和捕获接口信息等，有助于更全面地了解和分析网络通信。这种分层结构和信息组织使得 PCAP 文件成为网络分析和故障排查的重要工具，提供了详细的通信内容，有助于网络安全和性能优化。

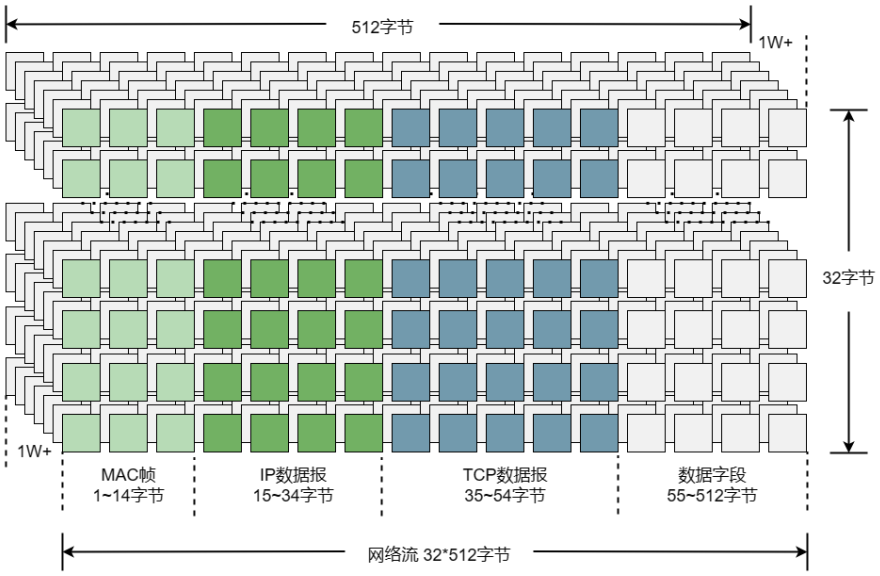

图 3.1 数据样本格式

3.4　数据集处理方法

在本次实验中，使用 Wireshark 工具从应用层或链路层开始捕获数据包。Wireshark 是一款广泛使用的网络协议分析工具，能够实时捕捉和显示网络上的数据包信息。它支持多种网络协议，并能够对数据包进行深入分析。为确保数据的完整性，捕获条件专门针对 SYN 和 ACK 报文，这两种报文在 TCP 连接建立和确认过程中起着关键作用。捕获的报文包括保留的（reserved）、乱序的（out-of-order）及重传的（retransmitted）报文，这些特性对于分析网络性能和诊断问题非常重要。实验结果是对九类不同应用的数据包（即 pcap 包）的收集。每个 pcap 文件都被输出到一个单独的目录中，便于后续处理和分析。

在网络数据包处理的流程中，首要步骤是借助于“get streams”函数。此函数通过遍历指定文件夹下的 JSON 文件，专注于对 TCP 和 UDP 数据包的解析，这两种是最为普遍的传输层协议。具体而言，该函数可以识别并提取网络流（stream）的相关信息，并巧妙地将这些信息存储在字典结构中。字典的采用使得信息可以迅速而方便地进行检索和更新。接下来，函数会将提取的十六进制数据转化成 32*512 的矩阵。这一步通常用于数据可视化或者进行更深层次的数据分析，例如使用机器学习方法来识别特定的网络行为模式。而生成的矩阵则会被保存为文本文件，以备进行后续处理。随后，该函数将处理结果写入名为“info.txt”的文本文件中。这个文本文件详细记录了数据包处理过程中的各类信息，如数据包类型、大小、时间戳等，对于理解网络流量特性和进行行为分析至关重要。

另一方面，“split pcap”函数则被用于处理庞大的 pcap 文件。鉴于 pcap 文件可能非常庞大，直接进行处理可能导致内存溢出或者处理速度过慢的问题。为了解决这一问题，该函数通过调用“editcap”工具，将大文件巧妙地分割成更小、更易于管理的部分。此外，“extract json”函数则利用“tshark”工具对 pcap 文件进行解析，并将其转换为 JSON 格式。JSON（JavaScript Object Notation）是一种轻量级的数据交换格式，易于人阅读和编写，同时也易于机器解析和生成。将 pcap 数据转换为 JSON 格式可以使得数据处理更加灵活和强大。

最后，“main”函数则充当整个处理流程的控制中心。该函数根据 pcap 文件的大小选择不同的处理方式，并调用上述函数对数据包进行处理，最终生成输出结果。整个代码通过日志记录执行过程，以方便进行跟踪和调试。日志记录在程序调试和维护中扮演着重要的角色，能够帮助开发者了解程序的运行状态，并快

速定位问题的源头。综合而言，本次实验通过 Wireshark 工具和一系列定制化函数，成功实现了对网络数据包的有效捕获、处理和分析，为网络性能评估、问题诊断和安全分析提供了强有力的工具和数据支持。

算法 1 Network Traffic Data Processing Algorithm

```
procedure MAIN(capture_path, save_folder, remove)
    Prepare and clean directories
    Initialize logger
    if size of capture_path < threshold then
        Call EXTRACTJSON
    else
        Call SPLITPCAP
        for each file in split folder do
            Call EXTRACTJSON
        end for
    end if
    Call GETSTREAMS
    if remove is True then
        Remove temporary directories
    end if
end procedure
procedure SPLITPCAP(capture_path, logger, save_folder)
    Run command to split pcap file into smaller files
end procedure
procedure EXTRACTJSON(capture_path, logger, save_folder)
    Run tshark command to extract data to JSON
end procedure
procedure GETSTREAMS(json_folder, logger, save_folder)
    Process JSON files to extract flow data
    Convert data to specified format and save
end procedure
```

3.5　本章小结

本章节我们首先介绍了 PCAP 相关概念和 PCAP 数据包内部的相关架构。然后我们分析了数据集的处理方法，并给出了相关伪代码。对于从传输层捕获到的数据包，通过传输层、网络层、链路层的报头结构计算出数据集中各个字段的物理含义。对于较大的数据包我们先对其进行了分割，最后我们将单个样本的结构一图像的形式进行了绘制。

第 4 章　基于深度学习的流量识别方法

本章节详细阐述了如何基于深度学习模型搭建神经网络框架进行流量识别任务，

4.1　研究动机

实验动机源于对传统流量识别方法的反思。传统方法通常依赖于手工设计的特征和规则，这在面对复杂的网络环境和新型攻击手法时显得力不从心。深度学习模型的引入为流量识别提供了一种强有力的解决方案。通过深度学习，模型可以自动学习数据中的抽象、高层次的特征，而不再依赖于人工定义的特征。这对于处理多样性、未知攻击模式和大规模网络流量的挑战至关重要。在深度学习方式的选择上，我们考虑了两种不同的处理方式，即基于图像处理的 CNN 和基于时序特征的 LSTM。鉴于使用图像处理的方式方便进行 SHAP（Shapley Additive exPlanations）特征值计算，我们最终选择了图像处理。这一选择不仅有助于提高解释性，还为后续的特征分析提供了更为方便的手段。对于图像处理，我们进一步选择了 ResNet 网络。这一选择考虑到了深度学习模型中常见的梯度消失或梯度爆炸问题，尤其是在卷积层过深的情况下。ResNet 以其创新性的残差学习结构成功解决了这一问题，为我们提供了更深层次网络结构的可能性。在流量识别中，我们经常面对高维度数据和复杂的特征关系，ResNet 的强大非线性建模能力使其成为一个理想的选择。通过引入残差块，ResNet 不仅能够更好地学习网络流量中的抽象表示，还提高了模型对于复杂网络流量模式的学习能力。在实验设计中，我们将首先进行流量数据的预处理，将原始网络流量数据转换成适合 ResNet 模型输入的图像形式。接着，我们将构建 ResNet 网络结构，选择适当的损失函数和优化器，并通过有监督学习对大规模的网络流量数据进行训练。在训练过程中，我们将关注模型的稳定性，通过正则化和批标准化等手段解决梯度消失和梯度爆炸的问题。最后，在评估阶段，我们将利用真实流量数据对模型进行性能评估，包括准确性、召回率、精确度等多个指标的全面考量。

综上所述，本实验旨在通过深度学习模型，特别是基于 ResNet 的图像处理方式，提高网络流量识别的准确性和泛化能力。通过深入研究深度学习在流量识

别中的应用，我们期望为网络安全领域的研究和实践带来新的思路和方法，使流量识别更加适应复杂多变的网络环境，提高网络安全水平。

4.2 深度学习中的流量识别方法

深度学习模型是人工智能领域的一个重要分支，主要基于人工神经网络的架构。这些模型通过模拟人脑的工作方式来处理和学习大量的数据。深度学习的成功部分得益于其能力在多层神经网络中自动提取和学习特征。深度学习模型通常由多层的神经网络构成，每一层包含若干神经元。这些神经元通过权重连接，能够处理输入数据并传递至下一层。深度学习模型基于人工神经网络（Artificial Neural Networks, ANN），这是一种模仿生物神经网络（比如大脑）工作机制的计算模型。神经网络包含多个层，每一层包含多个神经元或节点。这些层分为输入层、隐藏层和输出层。

当神经网络包含多个隐藏层时，我们称之为“深度”学习。每一层都会对输入数据进行转换和提取特征，层与层之间通过权重连接。它基于复杂的人工神经网络架构，特别是包含多个隐藏层的网络。这些多层网络能够学习数据的多级次抽象和表示，使得深度学习在图像和声音识别、自然语言处理等领域表现出色。深度学习模型通过反向传播算法来调整网络中的权重，这是一种高效的训练方法，可以最小化预测错误。深度学习的成功在很大程度上归功于三个关键因素：大量的训练数据、强大的计算能力、以及改进的训练算法。随着数据量的增加和计算能力的提升，深度学习模型能够学习更复杂的数据模式，解决以往算法难以处理的问题。

在未加密网络流量识别技术的领域中，深度学习模型已经显示出显著的潜力。这些模型特别适用于处理高维度和复杂的网络流量数据。例如，使用循环神经网络（RNN）来训练数据包序列，特别是基于卷积神经网络（CNN）和门控循环单元（GRU）的深度学习架构在序列分类方面表现出色。这些方法在处理加密流量时，相较于传统方法显示出更高的效率和准确性。

深度学习在流量识别中的应用主要包括流量分类、异常检测和流量预测等任务。特别地，在一个研究中，通过使用 10-Fold Cross Validation 方法，对不同的最小连接数阈值进行了测试，发现随机森林分类器在低阈值设置下仍然保持高于 85% 的准确率，而深度学习模型在包括随机森林在内的集成方法中表现最

佳。此外，研究还发现，对于具有更多潜在类别的输入，基线 RNN 表现出较高的偏差，而改进后的 CNN-RNN 结构则能有效提升准确度WiKibookBibliography。

在基于深度学习的未加密网络流量识别技术的研究中，除了 10-Fold Cross Validation，还采用了其他评估方法来验证模型的性能。例如，使用精确率、召回率和 F1 分数等指标来评估分类器的性能。这些评估指标提供了对模型在不同类别上的表现的全面视角，有助于理解模型的泛化能力和实际应用效果。特别地，在网络流量识别中，深度学习模型展现了在处理高维度和复杂数据方面的显著优势，尤其是在处理加密流量时，相比传统方法表现出更高的效率和准确度。这些研究结果表明深度学习技术在网络流量识别方面的巨大潜力和实际应用价值WiKibookBibliography。

随着网络技术的快速发展和数据流量的激增，网络流量识别成为网络管理和安全的关键环节。在此背景下，基于深度学习的未加密流量识别技术显得尤为重要。这些技术包括卷积神经网络（CNN）用于识别网络流量中的特定模式和行为，循环神经网络（RNN）适用于捕捉流量数据中的时间依赖性，自编码器用于异常检测，以及结合多种架构的混合模型以提高识别的准确性和鲁棒性。这些方法广泛应用于入侵检测、网络流量监控、服务质量（QoS）管理等多种网络安全和管理场景。尽管取得了显著进展，基于深度学习的未加密流量识别技术仍面临诸如数据集多样性和质量、模型可解释性和泛化能力、以及实时处理需求等挑战。未来研究可能集中于提高模型的精确度和效率，处理大规模流量数据，以及增强模型的可解释性和自适应能力，使其在网络流量的识别和分析方面发挥更大作用。总的来说，基于深度学习的未加密网络流量识别技术为网络安全和管理提供了有效的工具。随着技术的不断进步和深入研究，预期这些方法将在网络流量的识别和分析方面发挥越来越重要的作用。

本研究探讨了利用深度学习技术进行加密网络流量识别的有效性。研究主要集中在几种关键的技术类别上，包括循环神经网络（RNN）和其变体门控循环单元（GRU），用于处理数据包序列和时间序列数据；卷积神经网络（CNN）与 RNN 的结合，用于处理时间序列特征如到达时间间隔和有效载荷大小；集成方法，通过结合多个针对不同流量特征的分类器来提高整体识别准确性；以及自动化机器学习（AutoML）技术如 Auto-Sklearn，用于比较和自动选择最优的分类器。这些技术的融合使深度学习模型能够有效地识别加密流量的模式和行为特

征。

针对加密与未加密流量数据集的选择取决于项目的具体目标和实际需求。加密流量的优势在于它更符合现实世界的场景，尽管特征提取更为复杂，可能需要更长的训练时间，且准确率可能较低。另一方面，未加密流量的特征提取过程更直接，使得模型训练更快且准确率通常更高，但它可能不完全符合现实应用中的需求。如果目标是追求高准确率和快速开发，则未加密流量是合适的选择；然而，若重视模型在实际加密环境中的应用，尽管面临更大挑战，加密流量则是更实用的选择。

尽管 ResNet 这样的深度学习神经网络最初主要用于图像处理，但它们在加密流量识别方面也显示出了巨大潜力。考虑到加密流量数据的特点，我们可以将实验样本的 32x512 矩阵视为一张照片，其中每个单元格或“像素点”代表流量数据的一个特征。这样的视角为利用 ResNet 强大的图像处理能力提供了新的途径。通过分析每个像素点的 Shapley 值，可以有效地识别和解释哪些特征对流量识别起到了关键作用。这种方法允许我们利用 ResNet 网络对图像进行的深入特征提取和分析能力，将其应用于加密流量数据，以揭示流量模式和行为。因此，虽然 ResNet 需要进行适当的调整和优化以适应网络流量数据的特性，但其图像处理的本质使其成为处理加密流量识别挑战的有效工具。总体而言，这种方法不仅提供了一种新颖的视角来理解加密流量数据，还展示了深度学习技术在网络安全领域的广泛应用潜力。

4.3 实验设计

4.3.1 算法设计

在这个实验设计中，我们充分考虑了数据集准备的重要性，通过自定义数据集类 MyDataset（数据集采集处理方式见第三章），以清晰、结构化的方式组织数据，提高了数据管理效率和代码可维护性。这种设计能够为实验提供高质量、可靠的基础数据，确保了实验结果的稳定性。通过在数据集准备阶段的细致处理，我们能够更有效地利用实验资源，降低了因数据质量不佳导致的实验误差。在数据集划分方面，选择了 KFold 交叉验证方法，将数据集划分成了 5 个互斥的子集。这一创新性的划分方式为模型提供了更全面、更具代表性的训练和测试数据，减少了对特定数据划分的依赖性。这种交叉验证设计不仅提高了实验结果的

鲁棒性，还增强了模型对不同数据分布的适应能力。通过多次折叠中的数据轮流作为训练集和测试集，确保了实验评估的全方位性，使实验结果更为可信。在训练和测试的过程中，使用 DataLoader 实现了对数据的高效批量加载，这在大规模数据集上具有重要意义。选择 ResNet50 作为基础模型，结合交叉熵损失和带动量的 SGD 优化器，我们不仅充分考虑了模型结构的适应性，还注重了在训练过程中的有效性和稳定性。这样的设计保障了实验结果的可重复性和可比较性。在每个 epoch 中，通过对模型在训练集和测试集上的全面评估，包括准确率、精确率和召回率等指标，我们获得了对模型性能的深入理解。这有助于发现模型的潜在问题，并为进一步的调整和改进提供了有力的支持。通过这样的细致评估，我们能够及时发现模型在不同场景下的表现，使实验结果更为全面。在模型保存和指标可视化方面，设计了一个智能的保存机制，仅当模型在任何折叠的任何 epoch 中取得更高准确率时才保存。这种智能的保存策略确保了最终选择的模型在各个折叠中表现最佳，提高了实验结果的质量。同时，通过绘制图表展示训练损失、测试损失、准确率、精确率和召回率随 epoch 变化的趋势，使实验结果更加直观和可解释。整个实验设计的优点在于，通过合理的数据处理和模型训练设计，我们确保了实验结果的可靠性和高度可解释性。交叉验证的引入减小了对某一数据划分的依赖，使得实验结果更具普适性。通过高效的数据加载和合理的模型选择，我们提高了实验的效率和准确性。智能的模型保存机制和指标可视化进一步增强了对实验结果的监控和理解。

综合而言，我们通过以上设计内容实现了对深度学习模型在流量识别中的全面评估。采用 KFold 交叉验证、数据批量加载、ResNet50 网络等方法，保证了实验的科学性和可靠性。这一设计思路旨在提高模型的泛化能力、适应不同数据分布，为网络安全领域的流量识别研究提供了坚实的实验基础。

ResNet-50 网络结构的概览如下。网络开始于一个初始卷积层（conv1），使用 7×7 的卷积核，64 个输出通道，步长为 2，填充为 3，得到的输出尺寸为 $112 \times 112 \times 64$。接着是一个最大池化层（maxpool），其使用 3×3 的池化核，步长为 2，填充为 1，输出尺寸为 $56 \times 56 \times 64$。网络中的残差层由多个 Bottleneck 块构成，每个块包含了三个卷积层（1×1，3×3，1×1），扩展因子为 4。第一残差层（layer1）包含 3 个这样的块，输出尺寸保持为 $56 \times 56 \times 256$。第二残差层（layer2）包含 4 个块，输出尺寸减半至 $28 \times 28 \times 512$。第三残差层（layer3）包含 6

个块，输出尺寸为 $14\times14\times1024$。第四残差层（layer4）包含 3 个块，输出尺寸为 $7\times7\times2048$。网络以一个平均池化层（avgpool）结束，输出维度为 $1\times1\times2048$。最后，全连接层（fc）将特征图展平，并输出 9 个类别的预测。

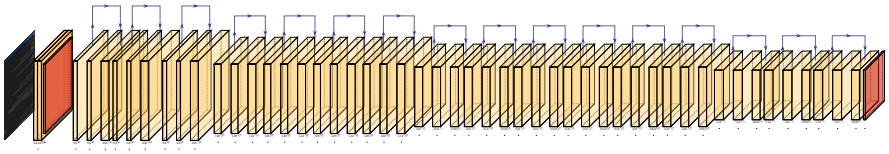

图 4.1 Resnet50 模型图

如图所示，Resnet50 网络主要包括初始处理（Stem 部分）、残差块（Residual Blocks）、一个池化层和一个全连接层。初始处理（Stem 部分）是网络的起始部分，是由一个 7x7 卷积（64 个输出通道，步长为 2，填充为 3）的初始卷积层，一个批量归一化层，一个 ReLU 激活层，最后是一个（步长为 2，填充为 1）3x3 的最大池化层组成。残差块（Residual Blocks）：ResNet-50 使用 Bottleneck 结构的残差块。每个 Bottleneck 块由三个卷积层组成——一个 1x1 卷积层用于降维，一个 3x3 卷积层为主要的卷积操作，再一个 1x1 卷积层用于升维。每个卷积层后都有批量归一化和 ReLU 激活。Bottleneck 块的特点是输出通道数是输入通道数的 4 倍。网络中有四层这样的残差块，分别为 layer1, layer2, layer3, layer4。这些层的块数分别是 3, 4, 6, 3。随着网络的深入，特征图的尺寸逐渐减小，而通道数增加。在网络的最后，使用了一个自适应平均池化层，将特征图的尺寸降至 1x1。最后是一个全连接层，通常用于 1000 类的 ImageNet 分类，其输出尺寸为 9（本实验是对 9 类不同的应用流量进行分类）。关于初始化和残差连接，网络使用 Kaiming 初始化方法。每个残差块的输出通过一个恒等映射（identity）与输入相加，从而实现残差连接。如果启用了 zero-init-residual，最后一个批量归一化层的权重会初始化为 0，这可以使残差分支以零开始，每个残差块表现得更像一个恒等映射。ResNet-50 的这种深层结构和残差连接有效地缓解了深度网络中的梯度消失问题，使得模型在训练过程中更加稳定，提高了模型的训练效率和准确率。

4.3.2 流量样本关键字段识别实验

这个实验的主要研究动机是为了确定在流量样本中，是报头字段（Header Fields）更加重要还是数据部分字段（Data Fields）更加重要。报头字段通常包含了关于数据包的元数据，如发送者、接收者、数据类型和协议信息等。这些信息

对于理解网络流量的行为模式和特征至关重要，因为它们提供了数据的来源和目的地，以及数据传输的基本方式。通过分析这些字段，可以有效地识别流量的类型和来源，这对于网络安全和流量管理尤为关键。另一方面，数据部分字段包含了实际传输的内容，这可能包括用户数据、应用程序数据等。这部分的分析可以揭示数据包的实际内容和目的，对于深度分析流量（如内容审查、数据泄露检测等）来说非常重要。这个实验的目的是通过系统地分析和比较报头字段和数据部分字段的重要性，来提高流量识别的效率和准确性，这对于网络安全和高效的网络流量管理具有重大意义。

为了评估报头字段和数据字段在流量识别中的相对重要性，我们设计了一个控制变量实验，即分别只保留报头部分和数据部分，同时将其他部分置为零。我们采用上一小节所介绍的 ResNet-50 模型对这两部分进行训练和评估。使用交叉验证和不同数据集的测试来确认通过 SHAP 值识别出的特征的有效性，确保关键信息不丢失。同时，充分测试模型在不同时间点和网络环境下的泛化能力和鲁棒性。通过这个控制变量实验，即分别只保留报头部分和数据部分，同时将其他部分置为零，我们可以分别观察这两部分的重要性。这样的方法可以帮助我们理解哪一部分对于流量识别来说更加关键。这不仅可以提高流量识别的准确性，也有助于设计更高效的数据处理算法，从而在网络安全、流量分析和管理中发挥重要作用。此外，这种方法还有助于优化资源分配，比如在资源有限的情况下，我们可以优先处理更重要的字段。

4.3.3　基于 Resnet50 的流量识别实验

在网络安全领域，深度学习在加密流量识别上的研究动机主要源于对识别精度的追求和对加密技术挑战的响应。随着网络流量量的增长以及加密技术的普及，传统的流量识别方法，例如基于端口号和负载内容的检测技术，已不再有效。深度学习模型，尤其是能从原始数据中自动学习特征的模型，如卷积神经网络 (CNN) 和循环神经网络 (RNN)（本次实验采用是的 RESNET 网络），已被证明在加密流量的分类和识别任务中具有较高的准确率。这些模型能够识别流量的统计特征和模式，即使在流量内容被加密的情况下，也能有效工作，从而无需解密就能进行有效的流量识别。此外，深度学习模型对于网络流量的多样性和动态性表现出了出色的适应性和泛化能力，这对于应对不断变化的网络威胁至关重要。实时处理能力的需求也促使研究者利用深度学习模型，因为这些模型可以部

署在高性能计算平台上，实现对流量的快速准确分析。总的来说，深度学习技术在加密流量识别领域的研究，旨在开发能够适应加密协议、自动化特征提取、以及满足实时监测需求的先进解决方案。

基于深度学习的未加密流量识别技术的研究动机主要源于对网络流量进行有效分类的需求。这种分类对于识别各种类型的流量（如合法、恶意或移动流量）至关重要。深度学习算法在实时识别方面显示出潜力，能够通过分析网络流的前几个数据包或字节来进行分析。此外，它们可以自动执行传统上手动进行的特征工程。使用卷积神经网络（例如本实验使用的 Resnet 网络）和门控循环单元（GRU）等深度学习架构进行序列分类是这一领域的重大进展。这些技术能够捕捉连续时间槽中特征向量之间的依赖关系，并适应性地处理可变长度的序列，这对于准确的流量识别至关重要。这种高级分类能力对于管理不断增长的网络流量的复杂性和量非常重要。

本实验旨在利用深度学习技术，尤其是 Resnet50 模型实现对应用流量的有效识别和分类。实验通过对应用层和传输层采集到的两份数据集进行流量分类从而形成对照实验。数据采集与处理如第三节所示，包括数据清洗、标准化处理以及将数据格式转换为模型可接受的形式。再实验的训练阶段，采用了交叉验证的方法以评估模型在不同数据集上的性能，确保实验结果的可靠性和模型的泛化能力。

4.4 测试方案

在进行深度神经网络 resnet50 进行流量识别实验后，我设计了一套综合而有效的测试方案，其中充分利用了 KFold 交叉验证作为主要的模型评估手段。KFold 交叉验证的独特之处在于将数据集分割成 K 个互斥的子集，通过在 K 次迭代中轮流将每个子集作为测试集，而其余的 K-1 个子集作为训练集，从而全面评估模型性能。这一方法有助于减少因特定数据集划分而引起的偏差，同时提高对模型泛化能力的评估。

特别值得强调的是，KFold 交叉验证在小数据集情境下展现出强大的优势，能够更有效地充分利用有限的数据资源。由于每个数据点都有机会作为测试数据，这种方法有助于减小模型评估的方差，提供更加鲁棒和全面的性能评估。因此，在流量识别实验中，KFold 交叉验证不仅可以确保实验结果的稳定性，还能

够增强模型在不同数据子集上的性能可靠性。通过这一综合测试方案，我们能够更深入地理解 resnet50 模型在网络流量识别任务上的表现，并提高对其性能的整体信心。

4.5　实验评估

我们对深度学习模型训练出来的效果在预留出的测试集上进行测试。我们所使用的是较新的 PyTorch v2.0.1 框架进行搭建；使用的 Docker 环境是 v20.10.10 版本，Python 采用的是 v3.10.12 版本，TensorFlow 采用的是 v2.13.0 版本。

4.5.1　度量指标

KFold 交叉验证是一种模型评估方法，它将数据集分割成 K 个子集。在 K 次的迭代中，每次选择一个子集作为测试集，而其余的 K-1 个子集作为训练集。这种方法有助于减少模型评估的偏差和方差，因为每个数据点都有机会作为测试数据使用。KFold 交叉验证特别适用于小数据集，能够更有效地利用有限的数据资源。

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Samples}} \tag{4.1}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4.2}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4.3}$$

训练损失是在训练过程中计算的，它表示模型在训练数据上的性能。通常，这是通过计算模型在训练集上每个样本的预测误差并求和得到的。测试损失则是在测试数据上计算的，它表示模型在未见数据上的性能。这两个指标都是评估模型拟合程度的重要工具。准确率（Accuracy）是最常用的性能指标之一，它简单地衡量模型预测正确的样本占总样本的比例。精确率（Precision）和召回率（Recall）则用于在二分类问题中更细致地评估模型性能。精确率是指模型正确预测为正类的样本数占模型预测为正类的总样本数的比例. 而召回率是指模型正确预测为正类的样本数占实际正类的总样本数的比例。在不平衡的数据集中，这两个指标尤其重要，因为它们可以帮助理解模型在预测不同类别时的性能。

4.5.2 实验结果

在流量样本关键字段识别实验中，我们开展了实验以评估在网络流量识别任务中报头字段和数据字段的重要性。具体来说，实验中分别将报头字段和数据字段置零，然后执行流量识别任务。实验结果显示，当保留报头字段时，识别准确率达到了 90%，而仅保留数据字段时，准确率下降到了 70%。这一差异突显了报头字段在流量识别中的显著重要性。通过对比实验中的两组柱状图——红色

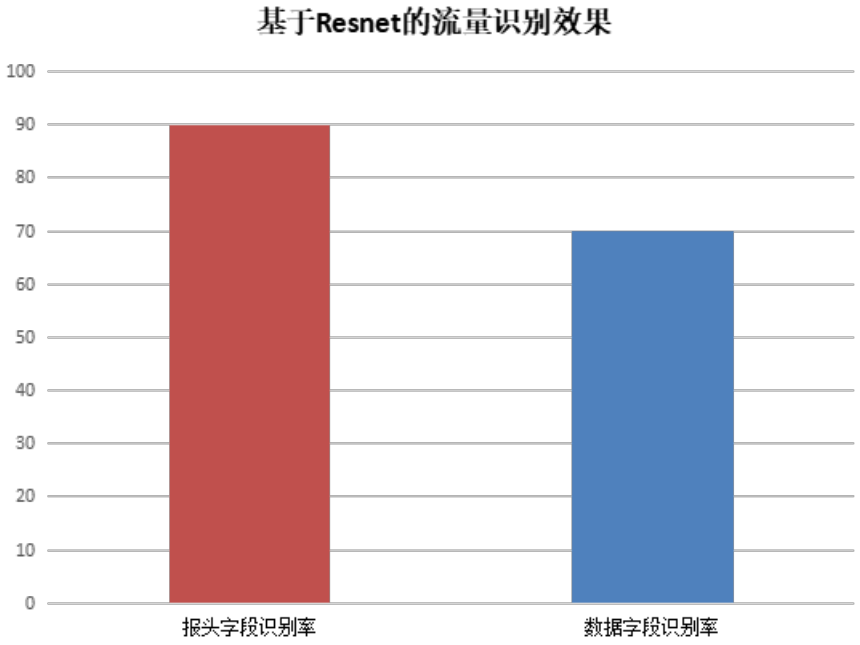

图 4.2 流量样本关键字段识别实验

柱状图代表报头字段，蓝色柱状图代表数据字段，我们可以直观地看出报头字段在流量识别中的贡献度高于数据字段。这可能是因为报头字段携带了数据包的路由和协议信息，这些信息对于判断流量的类型和来源是至关重要的。而数据字段尽管包含了传输的内容，但在识别流量类型阶段可能并不如报头信息重要。

根据提供的折线图，我们可以进行以下分析：训练损失（Training Loss per Epoch）：初始时损失较高，但在前几个 epoch 迅速下降，这表明模型开始时迅速从训练数据中学习。在之后的 epoch 中，训练损失波动较大，但总体趋势向下，这可能表示模型正在学习，但可能遇到了一些复杂的样本或者局部最优问题。测试损失（Test Loss per Epoch）：测试损失也表现出了快速下降，但在经过最初的学习阶段后，损失的下降速度减慢，并在后期呈现上升趋势。测试损失在后期的上升可能是过拟合的迹象，即模型可能过度适应了训练数据，而在测试数据上的

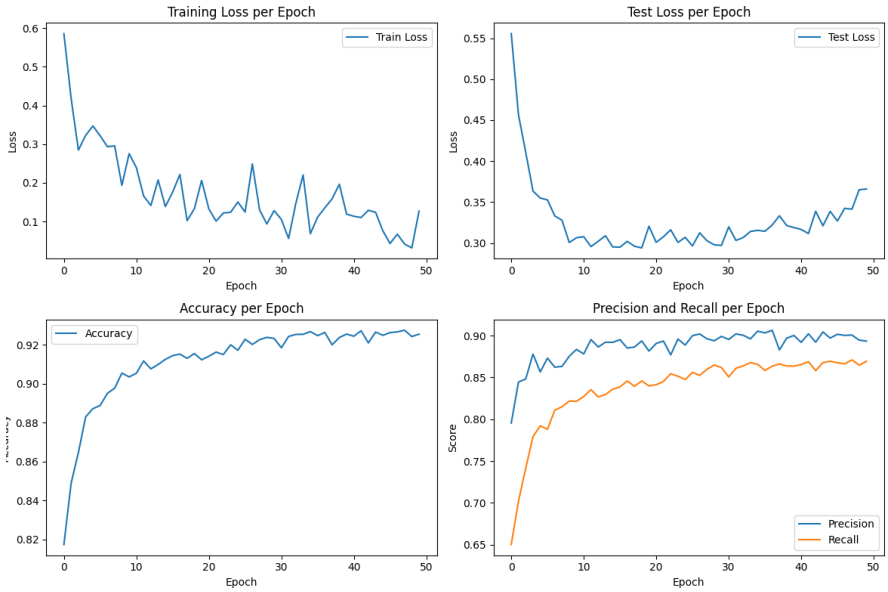

图 4.3 基于 Resnet50 的流量分类结果评估图

泛化能力下降。准确率（Accuracy per Epoch）：准确率快速上升，并在大约 20 个 epoch 后趋于稳定在 92% 左右。准确率的稳定性表明模型在训练集上具有一致的预测能力。精确度和召回率（Precision and Recall per Epoch）：精确度和召回率都随着 epoch 的增加而上升，但精确度的提高速度似乎快于召回率。在某些 epoch 之后，精确度和召回率都趋于平稳，其中精确度在百分之八十五附近，召回率在百分之七十五附近。这表明对于正例的预测，模型更倾向于保守预测（即更注重预测的正确性而不是预测所有正例）。根据这些观察结果，模型在训练过程中表现出了良好的学习能力，但测试损失的上升趋势和准确率的稳定表明可能需要进一步调整模型以避免过拟合，并可能需要探索不同的方法来提高召回率，尤其是如果模型的应用场景要求更高的召回率。可能的策略包括添加正则化、使用更多的数据进行训练、调整模型结构或超参数以及应用不同的数据增强技术。

4.6　讨论

在实验环境中网络流量识别领域，报头字段与数据字段各自承担着不同的角色，其重要性随着具体的应用场景和分析环境的变化而变化。报头字段，尤其在隐私保护至关重要的场合，发挥着至关重要的作用。由于这些字段通常不含敏感的有效载荷数据，它们成为了流量分类的核心。在加密流量分析中，报头字段更是成为了数据包中唯一可供观察的部分，包含了诸如协议类型、源和目的 IP 地址、端口号等元数据，这些信息是识别流量类型而无需深入有效载荷的关键。

另一方面，数据字段能够揭示流量内容的深层面貌，对于实现更细致的流量识别具有极高的价值，例如，它能帮助区分各种视频流服务。然而，随着通信加密技术的日益普及，数据字段的可见性和可用性在流量识别中受到了限制。这导致近年来，研究者和工程师越来越依赖于报头字段以及基于统计的流量特征分析，以实现有效的流量识别。在设计网络流量识别系统时，应优先关注报头字段。特别是在资源受限及安全领域的应用，如入侵检测系统（IDS）和恶意软件分析中，报头字段的分析对于快速识别和响应潜在威胁至关重要。报头字段的有效利用，尤其在隐私保护至关重要的场合，可以不涉及敏感的有效载荷数据，而在加密流量分析中，这些字段提供的元数据成为了唯一的可观察信息。为了提升流量识别的效率和准确性，需要对识别模型进行精细调优，选择合适的特征选择算法，调整模型参数以适应不同网络行为模式，并使用大规模且具有代表性的数据集。此外，实验设计的具体设置，如样本规模、报头与数据字段的详细信息，是确保实验成功和可靠结果解释的关键。一个结合多性能指标的综合性评估框架，有助于全面理解流量识别模型的效能和局限性。随着通信加密技术的发展，数据字段在流量识别中的可见性和可用性受到限制，促使研究者和工程师越来越多地依赖于报头字段以及基于统计的特征来识别流量，这对于深入理解流量的本质特征和动态行为模式至关重要。

4.7 本章小结

本章节我们首先分析了使用深度学习进行流量识别的动机和必要性，针对现有的流量识别问题，我们提出了基于 Resnet50 网络进行流量识别研究。我们根据现有的深度学习流量识别方法筛选出适合进行特征贡献度计算的神经网络。首先，我们采用控制变量法探索了报头字段和数据字段在流量识别中的相对重要性。然后，我们从应用层处理开放式系统互联模型（OSI）中的不同层中采集到的数据应用在了搭建好的神经网络中。并通过准确率等指标对识别效果进行了评估展示。实验表明相比于数据字段，报头字段更加有助于流量识别分类。例如在资源有限的情况下我们可以有限处理报头字段。其次，实验表明使用 Resnet50 网络对于流量识别任务具有高准确率的优点。

第 5 章　基于可解释性算法的数据轻量化方法

5.1　研究动机

数据轻量化技术是一种通过减小数据量的方法来提高模型性能和效率的技术。这种技术的主要目标是在保持模型性能的同时，降低模型对计算资源和存储空间的需求。这一领域的研究正在迅速发展，涵盖了多种技术和方法：首先，数据压缩技术是优化数据存储的一个主要方向。通过减少数据占用的存储空间，它不仅降低了存储成本，还可能提高数据处理速度，因为它减少了读写操作所需的时间。然而，这种方法可能会引入额外的处理开销，因为数据在使用前需要解压，这在某些情况下可能会降低效率。其次，数据清洗和预处理在确保数据质量方面发挥着关键作用。通过去除重复、错误或无关数据，这些技术提高了数据集的整体质量和可用性。然而，数据清洗和预处理可能非常耗时，特别是在大规模数据集上，它们可能需要复杂的算法和大量的计算资源。索引和数据分区也是提高查询效率的重要技术。它们通过组织数据结构来加快特定数据的检索速度。尽管如此，过度索引可能会导致维护成本的上升，并可能降低数据插入和更新的速度。并行处理和分布式系统的应用允许数据在多个处理单元上同时处理，显著提升了处理速度。这种方法尤其适用于大规模数据集和复杂计算任务。然而，它们也带来了复杂的数据同步和通信开销问题，可能需要精心设计的系统架构来优化。最后，机器学习和人工智能技术，如自动化数据标注和数据增强，也为数据集优化提供了新的视角。这些方法能够通过生成合成数据或改善数据标注的质量来增加数据集的多样性和质量。但是，这些技术可能依赖于大量的训练数据，并且可能受到算法的局限性。

我们需要一种低成本、高效率、影响小、泛化性强的数据轻量化方法。通过上述对数据轻量化技术优缺点的分析，我们可以使用模型重点关注的特征最为数据集进行神经网络训练。但是这种方法同样也面临着问题需要解决，即如何得到对模型分类启重要作用的特征序列。我们提出使用深度学习的可解释性算法进行计算模型关注的重要特征。深度学习模型的可解释性技术研究动机主要源自于对传统“黑箱”模型的透明度和可信度的追求。这些技术的目的是揭示模型的决策过程，以增加用户和监管机构的信任，同时提供更深入的洞察来指导模型

的调试和改进。重要的是，这些技术有助于确保人工智能的决策过程符合道德和法律标准，特别是在消除偏见和保证公平性方面。此外，可解释性技术使非技术专家能够理解和信任深度学习模型的预测，促进跨学科合作。在商业领域，这些技术还能帮助企业展示其产品的有效性和可靠性，从而提升客户信任和商业价值。总而言之，深度学习模型的可解释性技术是确保人工智能技术可持续发展的关键因素，它不仅提高了模型的透明度和可靠性，还增强了模型在各个行业中的应用价值。

5.2 可解释性相关概念

可解释性技术（Explainable AI，简称 XAI）旨在使复杂的机器学习模型变得更加透明，让用户能够理解、信任并有效管理 AI 系统。这些技术帮助揭示了模型的决策过程，使我们能够理解模型为什么会做出特定的预测。在机器学习模型的设计者本身也无法解释为什么人工智能能达到某些成果的情况下可解释性技术显得尤为重要。可解释性技术的发展被推动了由需求：在代理人之间的合作中（代理人指的是算法与人），信任至关重要，而可解释性技术就是建立这种信任的桥梁。例如，人工智能在图像识别中的应用，有时会学到一些无法解释的技巧，如利用图片上的版权标记来识别马的图片（图 2.2），这显然不是期望的行为。可解释性技术项目的目标是创建既具有可解释性又能保持 AI 性能的“玻璃盒”模型。在监管法令方面，官方机构和普通用户需要清晰的决策过程和规则来确保 AI 的可信任性和透明度。例如，欧盟一般数据保护条例（GDPR）提出了“要求解释的权利”，旨在解决算法可能带来的问题。在实现可解释性方面，有几种技术可以用来提高模型的可解释性。例如，全局可解释性关注模型如何基于整个特征空间进行决策，而局部可解释性则侧重于单个样本的预测解释。还有一些与模型无关的方法，如使用易于理解的模型（线性回归、决策树等）或者计算特征重要性的算法。Permutation Feature Importance 是一种评估特征重要性的方法，通过随机打乱特征值并观察对模型性能的影响来评估。而 Partial Dependency Plots（PDP）则可以展示特征值和预测结果之间的关系，帮助理解不同特征值如何影响模型输出。

深度神经网络的内部解释性：一项关于解释深度神经网络内部结构的综合调查概述了这一领域的关键进展和挑战。调查将解释性方法分类，基于它们解释

DNN 的哪一部分（权重、神经元、子网络或潜在表示）以及它们是在训练期间（内在）还是训练后（事后）实施的。它强调了几种技术，例如自解释模型、对抗性训练和去耦合，使 AI 系统更具解释性。这些方法旨在以人类可理解的术语提供解释，这是建立信任和诊断 AI 系统潜在失败的关键方面。（图 2.1）WiKibook Bibliography。

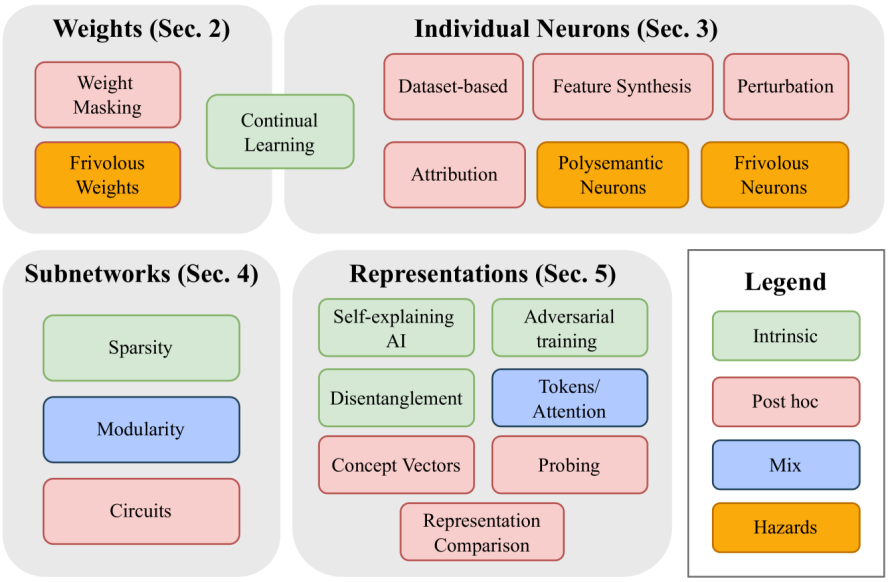

图 5.1 深度神经网络的内部解释性

概念相关性传播（CRP）：Fraunhofer Heinrich-Hertz-Institut 和柏林学习和数据基础研究所的最新工作导致了概念相关性传播（CRP）的开发。CRP 是一种最先进的方法，可以以人类可理解的概念解释 AI 决策。它不仅识别输入中的相关特征，还解释了这些概念的基础，它们在输入中的表示，以及负责这些概念的神经网络部分。这种方法在 AI 解释性方面标志着重大进步，全面审视了从输入到输出的过程，并为 AI 评估和互动设定了新标准。WiKibook Bibliography。

可解释人工智能（XAI）的增长：AI 模型的日益复杂化突显了可解释 AI 的需求。XAI 旨在使算法输出的原理对人类可理解。这个领域使用数学技术来检查 AI 模型中的模式，得出它们的决策过程的结论。美国国家标准技术研究所概述了 XAI 的四项原则：为所有输出提供证据，确保解释对用户来说是可理解的，确保解释真实反映了到达该输出的过程，以及系统只在其设计条件下或当系统对其输出有足够的信心时运行。XAI 的发展对于解决 AI 偏见和公平问题至关重要，因为模糊和有偏见的算法可能在各个领域（如刑事司法、社会服务和医疗保健）产生深远的影响。WiKibook Bibliography。

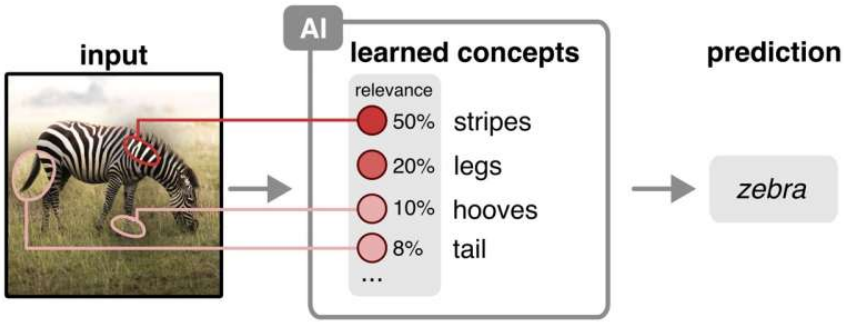

图 5.2 人工只能可解释性 (CRP)

在机器学习和深度学习的快速发展过程中，模型的可解释性成为了一个重要的研究领域。随着这些模型在现实世界的广泛应用，了解它们的内部工作机制和决策过程变得至关重要。本文综述并分类了当前的主要可解释性技术。基于最新研究，我们可以将机器学习和深度学习的可解释性技术大致分为以下几类：

内在解释性技术：这类技术通过对模型结构的深入理解来增强解释性。例如，自解释模型（Self-Explaining Models）这些模型在设计时考虑了解释性，如通过内部生成解释性的输出来直接解释其决策过程。对抗性训练（Adversarial Training）侧重于模型内部的特定组件，如权重、神经元或子网络，以提供直观的解释。通过对模型进行对抗性训练，增强其在各种情况下的解释性，同时提高模型对抗性攻击的鲁棒性。WiKibook Bibliography。

事后解释性技术：事后解释性技术旨在对已训练好的模型进行分析，以揭示其决策逻辑。例如，概念相关性传播（CRP）通过识别决策过程中重要的输入特征和概念，来解释模型的决策。WiKibook Bibliography。

可视化和特征归因技术：这类技术通过可视化方法（如热图）或特征归因来增强模型的可解释性。例如，层级相关性传播（LRP）通过突出显示对决策过程重要的输入区域，帮助理解模型的工作方式。梯度加权类激活映射（Grad-CAM）利用梯度信息来产生可视化的类激活热图，解释卷积网络的决策。WiKibook Bibliography。

基于规则的方法：这类方法通过提取模型的决策逻辑为简单的规则，使非专家用户也能理解模型的工作原理。例如决策树通过将模型的决策过程转化为一系列简单的规则和决策路径来提供解释。规则提取技术如 Skope-rules，通过提取简单规则来解释复杂模型。

基于模型简化的方法：这类方法通过创建简化版的模型（如线性模型或决策树）来近似原始复杂模型，以提供更易于理解的解释。线性模型近似：例如 LIME（局部可解释模型-敏感解释），通过在局部使用简单模型来近似复杂模型的行为。简化决策树：如 TreeInterpreter，通过生成简化的决策树来近似复杂模型。

5.3　SHAP 相关概念概述

Walls 等 (2013) 根据 Betts 等 (2005) 引用示例在现代机器学习领域，模型的可解释性越来越受到重视。尤其在高度复杂的模型如深度学习中，解释每个特征对模型预测的影响成为一个关键的挑战。SHAP（SHapley Additive exPlanations）作为一种先进的解释方法，为我们提供了深入理解模型决策过程的工具。SHAP 基于博弈论中的沙普利值（Shapley Value），这是一种公平分配合作游戏收益的方法。在机器学习的上下文中，沙普利值用于量化每个特征对模型预测的贡献。其核心思想是将一个预测问题转化为一个合作游戏，其中每个特征都被视为一个“玩家”，共同“合作”达成最终的预测结果。

沙普利值的计算方式：在机器学习模型中，每个特征被视为一个”玩家“，计算时需要考虑包括特征的所有可能子集。对于每一个特征组合，考虑当该特征存在或不存在时的情况。对于每个特征，计算其在所有特征子集中存在与不存在时对模型预测的影响。这种影响被视为该特征对预测的边际贡献。沙普利值是所有可能特征子集上的边际贡献的加权平均值。这个权重取决于特征子集的大小和特征总数。沙普利值确保每个特征的贡献公平，考虑了它与其他特征的所有可能组合。（Shapley Value 计算公式如下）

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [v(S \cup \{i\}) - v(S)] \tag{5.1}$$

其中：

- N 是所有特征的集合。
- S 是不包含特征 i 的特征子集。
- $v(S)$ 是特征集 S 对预测结果的贡献。
- $|S|$ 是集合 S 的大小。
- $|N|$ 是特征总数。
- $v(S \cup \{i\}) - v(S)$ 表示当包含特征 i 时相比不包含它时预测的变化。

SHAP 值的计算涉及到考虑所有可能的特征子集，并计算在包含或不包含特定特征的情况下模型预测的变化。这种方法通过考虑所有可能的特征组合，计算每个特征的平均边际贡献。公式上，SHAP 值是特征子集的边际贡献的加权平均。一个显著的优点是 SHAP 的模型不可知性（model-agnostic）特性。这意味着它可以应用于任何类型的机器学习模型，无论是简单的线性回归还是复杂的深度神经网络。这种通用性使得 SHAP 成为跨不同模型类型的可解释性分析的强大工具。

SHAP 不仅提供单个预测的局部解释，还能为模型的整体行为提供洞察。通过分析多个预测的 SHAP 值，我们可以理解模型在整体上是如何响应不同的特征组合的。这有助于揭示模型的行为模式，以及潜在的偏差或不公平性。

总而言之，SHAP 提供了一个强大的框架，用于解释和理解机器学习模型的预测。通过提供精确的特征级解释，SHAP 有助于提高模型的透明度和可信度。对于那些需要对模型的决策过程进行严格审查的应用场景，如医疗诊断、金融风险评估等，SHAP 的重要性不言而喻。未来，随着人工智能的进一步发展，我们预计 SHAP 将在解释模型预测方面发挥更加关键的作用。

5.4 算法设计

我们采集了尺寸为 32×512 的矩阵样本数据作为训练输入。为了明确每个数据点（或“像素”）在模型预测中的作用，我们使用了 SHAP 方法。首先，将每个样本视作 1×32×512 的图像，并针对每个像素计算其 SHAP 值，代表该像素的预测贡献度。当处理测试集时，同样为每个样本计算 SHAP 值，并对所有样本的结果进行位置对应的累加，形成一个汇总的 SHAP 值矩阵。为了洞察特征的重要性，我们将此矩阵中每一列的 SHAP 值进行累加，得到 1×512 的向量，其中每个元素均代表一个特征在预测中的总贡献度。基于这些贡献度，我们最终对特征进行了排序，以识别其对模型预测的重要性。

为了深入探讨模型决策过程中的各个特征的重要性，我们采用了以下步骤，具体如下图所示：

数据获取：首先，我们收集了样本数据，其中每个样本是一个 32×512 的矩阵。这些矩阵随后被用作训练输入。

计算单个样本的 SHAP 值：每个样本矩阵可以视作一个尺寸为 1×32×512 的

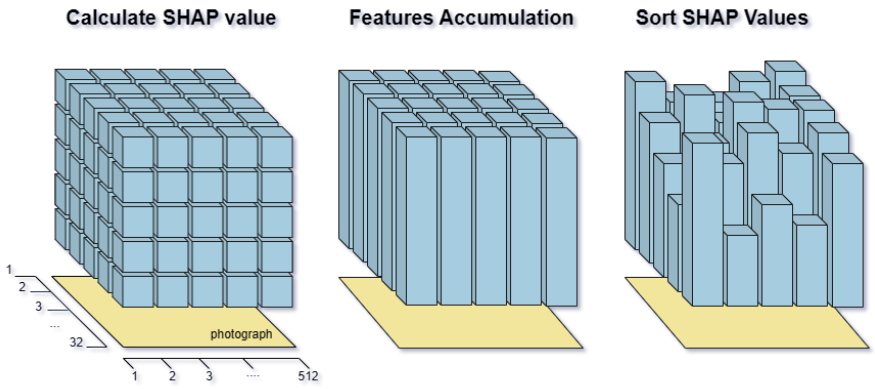

图 5.3 特征贡献度计算过程

图像。为了理解每个特定“像素”对输出结果的影响，我们计算了每个像素位置的 SHAP 值，从而得到该像素对最终预测的贡献度。

计算测试集的 SHAP 值：对于测试集中的每一个样本，我们都按照上述方式计算其 SHAP 值，生成了一个相同的尺寸为 1×32×512 的矩阵。随后，我们将这些矩阵在对应的位置上进行累加，从而得到了一个总结了所有样本的累加 SHAP 值的矩阵。特征排序：我们进一步加工得到的 SHAP 值矩阵，以理解各个特征的相对重要性。为此，我们将每列的 SHAP 值进行累加，从而得到一个 1×512 的矩阵，其中每个值代表了在测试集上，该特征对预测结果的总贡献度。最后，我们根据这些累计贡献度对所有特征进行排序，以确定其重要性。

shap.GradientExplaine 是 SHAP（SHapley Additive exPlanations）工具中的一个类，用于通过预期梯度方法来解释模型。这种方法是整合梯度方法的扩展，整合梯度方法是 Sundararajan 等人在 2017 年提出的，用于可微分模型的特征归因方法。它基于将沙普利值（Shapley values）扩展到无限玩家游戏的奥曼-沙普利值（Aumann-Shapley values）。

整合梯度值与 SHAP 值的主要区别在于，整合梯度需要一个单一的参考值来进行整合。预期梯度通过将积分重新表述为期望，并将该期望与从背景数据集中抽取参考值的样本相结合，来适应这一点。这导致了梯度的单一组合期望，这些期望收敛于归因于模型预期输出与当前输出之间差异的总和。

这种方法特别适用于深度学习模型，其中理解每个输入特征对模型预测的贡献至关重要。该方法允许对模型行为进行更细致的解释，特别是在多个特征以非线性方式交互的复杂场景中。

5.5 测试方案

WiKibookBibliography 尽管验证机器学习和深度学习模型的可解释性是一项复杂的任务，尤其是因为定义和衡量可解释性本身存在困难，但 Doshi-Velez 和 Kim 在 2017 年提出的框架为此提供了一个有用的指南。根据他们的研究，评估模型可解释性的方法主要分为三种：应用基础评估、人为基础评估和功能基础评估。应用基础评估涉及对专业人员执行特定实际任务的用户测试，重点在于评估模型或可解释性系统在实际任务中提供的实际帮助，并以某种理想化的绩效为衡量标准。人为基础评估则在人为设置和简化的任务中进行，这减少了时间和资源成本，并允许在更受控制的环境中测试更抽象的概念，例如在时间限制下解释的有效性。最后，功能基础评估不涉及人类受试者，而是利用可解释性的代理测量（如稀疏性、局部忠实度等），在多个数据集或应用程序上评估该方法的效果。这些不同的方法提供了一套全面的工具，用于评估和理解机器学习和深度学习模型的可解释性，帮助研究者和实践者从多个角度分析和改进他们的模型。

在本实验项目中，我们致力于通过网络流量分析与深度学习相结合的方法来识别关键特征，以优化数据采集过程。特别地，我们采用 SHAP 值来确定对模型预测至关重要的特征，并在保留这些关键特征的同时，将其他特征置零，以形成一个 32×512 的矩阵。在这一过程中，我们注重保证关键信息的完整性，这通过交叉验证和在不同数据集上的测试来确认，以确保这些特征确实对模型的预测具有决定性影响。

我们还注意到，在网络流量数据分析中，将非重要特征置零可能会导致信息的丢失，因为在特定情况下，某些看似不重要的特征可能成为关键。因此，在将优化后的模型投入实际应用之前，我们将其与全特征模型进行对比，评估它们在多个数据集上的性能。这不仅包括准确度，还包括模型的泛化能力和鲁棒性。我们进行了广泛的测试，包括时间泛化测试（即使用未来数据测试模型）和在不同网络环境下的测试。

为了全面评估模型性能，我们采用了准确度、损失值、混淆矩阵以及精确度和召回率等评估指标。这些指标帮助我们深入理解模型在不同类别上的表现，尤其是在类别不平衡的数据集中。

在模型训练与评估方面，我们延用了上一节介绍过的 ResNet 模型。模型通过 sklearn 的 KFold 进行了 K 折交叉验证，这是一种评估模型在数据集上性能的

强健方法。我们实施了标准的训练循环，包括前向传播、损失计算、反向传播和参数更新。评估循环则涵盖了损失、准确率、精确度和召回率的计算，这些都是分类任务中的关键指标。

5.6　实验评估

我们所使用的是较新的 PyTorch v2.0.1 框架进行搭建；使用的 Docker 环境是 v20.10.10 版本，Python 采用的是 v3.10.12 版本，TensorFlow 采用的是 v2.13.0 版本。

5.6.1　度量指标

KFold 交叉验证是一种模型评估方法，它将数据集分割成 K 个子集。在 K 次的迭代中，每次选择一个子集作为测试集，而其余的 K-1 个子集作为训练集。这种方法有助于减少模型评估的偏差和方差，因为每个数据点都有机会作为测试数据使用。KFold 交叉验证特别适用于小数据集，能够更有效地利用有限的数据资源。

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Samples}} \tag{5.2}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{5.3}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{5.4}$$

训练损失是在训练过程中计算的，它表示模型在训练数据上的性能。通常，这是通过计算模型在训练集上每个样本的预测误差并求和得到的。测试损失则是在测试数据上计算的，它表示模型在未见数据上的性能。这两个指标都是评估模型拟合程度的重要工具。准确率（Accuracy）是最常用的性能指标之一，它简单地衡量模型预测正确的样本占总样本的比例。精确率（Precision）和召回率（Recall）则用于在二分类问题中更细致地评估模型性能。精确率是指模型正确预测为正类的样本数占模型预测为正类的总样本数的比例. 而召回率是指模型正确预测为正类的样本数占实际正类的总样本数的比例。在不平衡的数据集中，这两个指标尤其重要，因为它们可以帮助理解模型在预测不同类别时的性能。

5.6.2 实验结果

本实验采用了 ResNet 网络对网络流量数据进行深度学习训练，并且通过计算每个特征的 Shapley 值来评估它们在流量识别中的重要性。实验结果表明，关键的报头字段包括 IP 首部校验、TCP16 位源端口号、TCP16 位窗口大小、TCP32 位确认序号、MAC 目的地址、IP 数据包总长度、TCP16 位目的端口号、TCP32 位序列号、IP 标识和 TCP 选项字段等。这些特征根据其 Shapley 值的大小被识别为在流量识别中至关重要的字段。

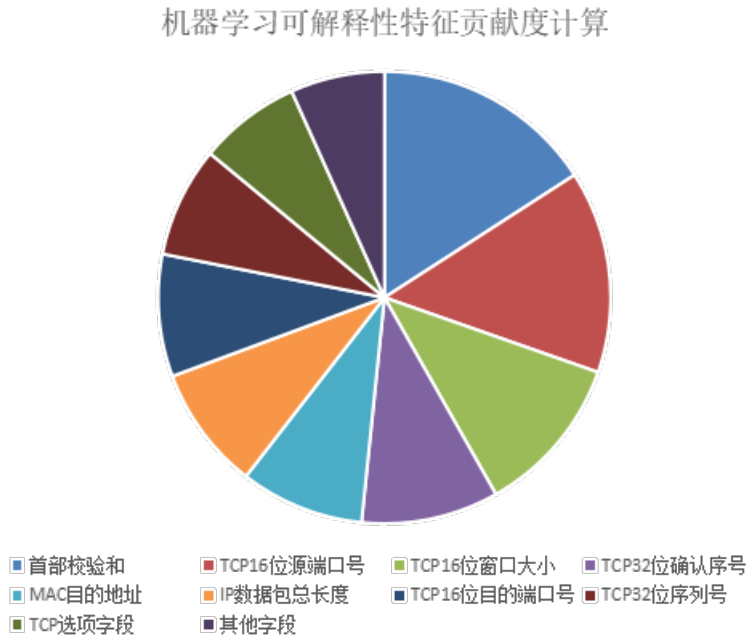

图 5.4 基于 Resnet50 的流量分类结果评估图

训练损失（Training Loss per Epoch）：训练损失在初始几个 epoch 内显著下降，这表明模型在训练的初期阶段迅速从数据中学习。在下降的初期之后，训练损失基本保持平稳，这可能意味着模型达到了一个相对较好的稳定状态。

测试损失（Test Loss per Epoch）：测试损失在最初的几个 epoch 后迅速上升至一个非常高的值，这可能表明在某些 epoch 中模型在测试集上表现非常差，这通常是过拟合的一个迹象。然而，之后测试损失迅速下降，并且在余下的 epoch 中保持较低的水平。这种下降和平稳可能表示模型在测试集上也学到了有用的泛化信息。

准确率（Accuracy per Epoch）：准确率在初始几个 epoch 迅速上升至 1.0 或 100%，这表明模型在训练数据上实现了完美的分类。通常，模型准确率达到 100%

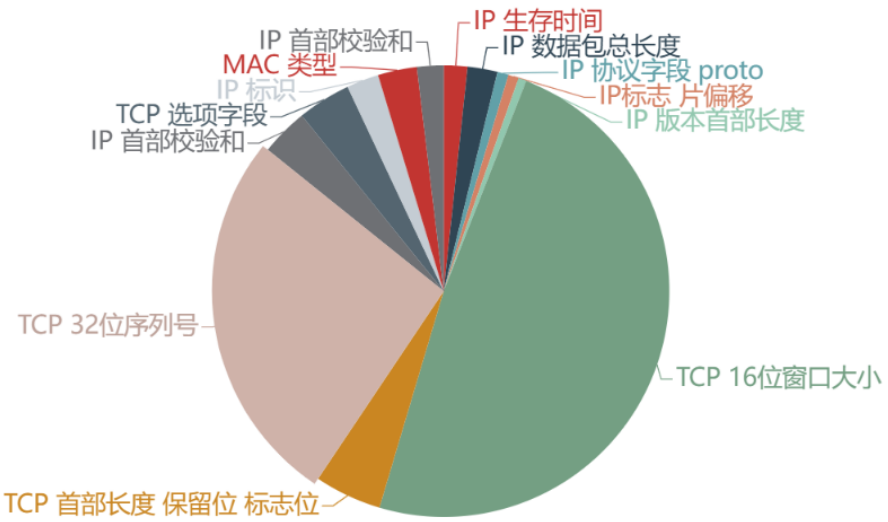

图 5.5 shap 计算结果

是过拟合的一个标志，尤其是如果测试损失没有显示相似的性能。

精确度和召回率（Precision and Recall per Epoch）：精确度和召回率在初始几个 epoch 后迅速上升至 1.0 或 100%，这与准确率的表现一致。同样，这样高的精确度和召回率通常不太可能反映真实的泛化性能，尤其是当测试损失图显示异常波动时。

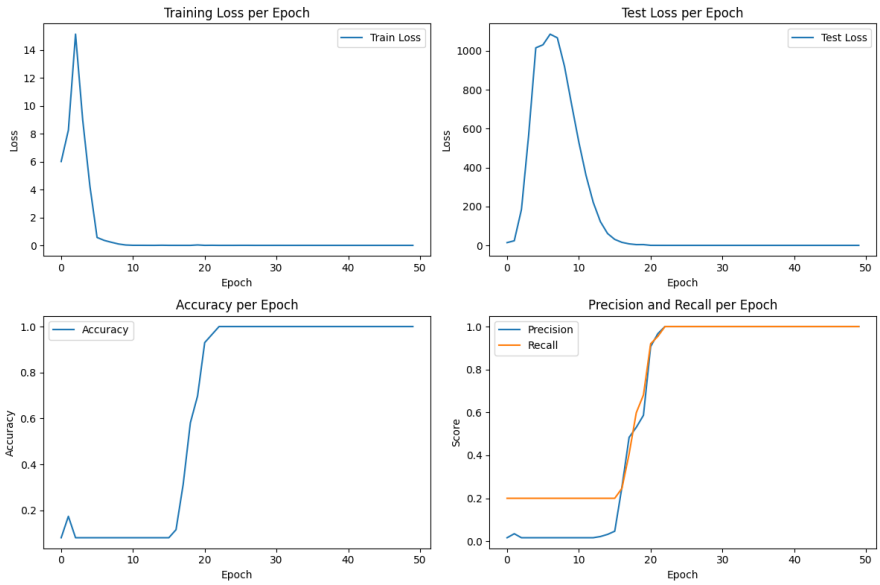

图 5.6 基于 Resnet50 重要特征的流量分类结果评估图

综上所述，这些图表显示的指标可能不完全准确反映了模型的实际性能。尤其是测试损失的异常波动和其他指标达到完美表现的情况表明，可能存在数据泄露、错误的评估协议或日志记录问题。需要进一步检查数据处理流程、评估方

法和日志记录系统，以确保这些指标是准确和可信的。此外，考虑到准确率、精确度和召回率都达到了 100%，还应该检查是否存在过拟合，并采取相应的措施（如交叉验证、正则化、或引入更多的训练数据）以提高模型的泛化能力。

5.7 讨论

在网络流量识别中，尤其是使用 IP 地址和端口号作为关键识别指标的情况下，存在多重挑战。动态 IP 分配、网络地址转换（NAT）和端口重用都增加了识别的复杂性和不确定性，使得基于 IP 地址和端口号作为主要识别依据变得不可靠。在这种背景下，基于深度学习模型处理未加密流量逐渐展现出潜力。这些模型能够从原始数据中自动提取关键特征，并能适应网络流量的多样性和动态性。因此，深度学习模型在未加密流量的识别中可能是一个更优的选择，因为它们可以访问和利用更完整的流量内容，减少对特定字段的依赖，并提供更快的实时处理能力。

总体而言，深度学习模型 Resnet50 为网络流量识别提供了一个强大的工具，特别是在面对未加密流量时。这些模型可以提高识别的准确性和效率，并能适应加密协议、自动化特征提取以及满足实时监测的需求。然而，在实际应用中，选择最合适的流量识别方法应基于具体的网络环境和需求。

5.7.1 TCP 16 位窗口大小

这个参数反映了接收端可以接受的数据量，影响着数据传输的速率和网络的拥塞状况。在分类应用流量时，不同类型的应用（如视频流媒体、文件下载、网页浏览）可能会有不同的窗口大小特征，这有助于区分它们。视频流和普通网页浏览流量在 TCP 16 位窗口大小方面通常有一些显著的差异：

视频流的 TCP 窗口大小：较大：视频流，特别是高清或实时视频，需要较大的带宽和更连续的数据传输。因此，它们通常会有一个较大的 TCP 窗口大小，以便能够更快地接收更多的数据。变化性：为了适应不同的网络条件，如带宽变化或拥塞，视频流可能会动态调整其 TCP 窗口大小。普通网页浏览的 TCP 窗口大小：

较小：相比于视频流，普通的网页浏览（如阅读网页内容、查看图片等）通常需要较少的数据量。因此，其 TCP 窗口大小通常会较小，因为这类流量较少且不连续。稳定性：网页浏览在 TCP 窗口大小上通常比视频流更稳定，因为网

页数据的传输需求相对固定，不像视频流那样频繁地需要调整。

总体而言，视频流由于其对高带宽和流畅性的需求，通常会使用更大的 TCP 窗口大小（PPTV、aiqiyi），而普通网页浏览则倾向于使用较小且更稳定的窗口大小。这些特征在使用机器学习模型对网络流量进行分类时非常有用，因为它们可以帮助模型区分不同类型的流量。

5.7.2　TCP 首部长度、保留位、标志位

TCP 首部的这些信息反映了 TCP 协议的具体使用方式。例如，不同的标志位组合（如 SYN、ACK、FIN）可以表明不同类型的网络行为（如连接建立、维持、终止），这有助于区分不同应用的流量特征。TCP 首部长度、保留位、和标志位是 TCP 协议中重要的字段，它们在区分不同应用的流量特征中扮演着关键角色：

TCP 首部长度：这个字段指示 TCP 头部的大小，可以间接反映出正在使用的 TCP 选项。不同的应用可能会使用不同的 TCP 选项集，从而影响头部长度。例如，一些高级应用（如某些数据库应用或定制协议）可能使用较多的 TCP 选项，从而导致更大的首部长度。

TCP 保留位：这些位在标准的 TCP/IP 协议中未被使用，但有时会被某些特殊的应用或协议扩展使用。在大多数常规应用中，这些位通常是未设置的。如果检测到这些位被设置，这可能表明流量来自于使用非标准或定制 TCP 扩展的应用。

TCP 标志位：这些位非常关键，包括 URG（紧急）、ACK（确认）、PSH（推送）、RST（重置）、SYN（同步）和 FIN（结束）等。SYN 和 FIN：SYN 通常用于初始化 TCP 连接，而 FIN 用于关闭连接。一次完整的 TCP 连接通常开始于 SYN 标志，结束于 FIN 标志。PSH：推送标志可能表明应用试图快速发送数据，这在需要实时通信的应用（如 VoIP 或某些游戏）中更常见。RST：如果频繁出现 RST 标志，可能表明连接问题或非正常的连接终止，这可能与特定类型的应用或网络行为有关。ACK：几乎在所有 TCP 数据传输中都存在，用于确认收到的数据。

通过分析这些字段的组合和模式，可以推断出不同应用的流量特征。例如，频繁的 PSH 标志可能与实时音视频流相关 (kugou 音乐)，而 SYN 和 FIN 的正常

模式可能指示常规的 Web 浏览活动。通过机器学习模型对这些特征进行训练和分析，可以有效地区分和识别不同应用产生的流量。

5.7.3 TCP 的 32 位序列号和 32 位确认序列号

序列号用于保证数据传输的顺序性和完整性。在流量分类中，序列号的模式可以帮助识别不同的应用行为，如大量连续的数据传输可能表明文件下载或视频流等。TCP 的 32 位序列号和 32 位确认序列号是 TCP 协议中非常关键的部分，它们在不同应用的通信行为中扮演着重要角色。虽然序列号和确认号本身不直接揭示特定应用的信息，但它们的模式和使用方式可以与某些应用的通信行为相关联：大量数据传输应用：如文件下载、视频流媒体服务或大型数据备份。这些应用通常会有一个连续增长的序列号模式，因为它们涉及到大量连续数据包的传输。确认序列号通常会快速增长，反映出高速数据接收。

实时通信应用：如 VoIP（语音通信）、实时视频会议或在线游戏。这些应用可能显示出较频繁的序列号和确认号交换，以保持实时数据流的连续性和稳定性。序列号的增长可能不像大量数据传输那么快速，但更为频繁和规律。

交互式应用：如网页浏览、即时消息或电子邮件。在这些应用中，序列号和确认号的增长可能较慢且不连续，因为数据传输通常是按需进行的。用户的交互（如点击链接、发送消息）会导致序列号和确认号的变化。

短连接应用：如 API 调用或某些数据库查询。这些应用通常涉及短暂的连接，每个连接只传输少量数据。序列号和确认号可能只在短时间内有较小的变化。

重试或连接问题：如果观察到序列号重复或确认号停滞不前，可能表明数据包丢失或连接问题，需要重传数据。这种模式可能与网络质量问题或具有重试机制的应用相关。

通过分析 TCP 序列号和确认号的变化模式，可以对应用的通信行为进行一定程度的推断。然而，这种分析通常需要结合其他网络特征和上下文信息，才能更准确地识别特定应用。

5.7.4 IP 首部校验和

实际上，IP 首部校验和并不直接反映不同应用产生的流量类型。IP 首部校验和是一个网络层的功能，用于确保 IP 头部在传输过程中的完整性和正确性。

它是一个计算出的值，用于检测数据在传输过程中是否发生了错误或变化。

对于不同的应用来说，它们产生的流量的 IP 首部校验和值并不是一个区分应用类型的可靠指标。这是因为：

标准化计算：IP 首部校验和是按照标准化的算法计算的，这个算法对于所有的 IP 数据包都是一样的。它的值完全取决于 IP 头部的内容（如源 IP 地址、目的 IP 地址、协议类型等），而这些内容与应用类型没有直接关联。

传输层以上的信息不影响：IP 首部校验和仅仅检查和保护 IP 头部的信息，它不涉及传输层（如 TCP 或 UDP）或更高层次（如应用层数据）的信息。因此，不同应用的流量在 IP 层面上可能非常相似。

目的是错误检测：校验和的主要目的是检测在传输过程中可能发生的错误，而不是用于区分或识别不同的应用类型。

总而言之，IP 首部校验和并不适用于区分不同应用产生的流量。要准确地识别和分类网络流量，通常需要考虑更多层面的信息，包括传输层协议、应用层数据模式、连接行为等。

5.7.5　TCP 选项字段

TCP 选项字段在分类应用流量时起到了重要作用，因为不同类型的应用在使用 TCP 协议时可能会根据其需求和特性使用不同的 TCP 选项。这些选项提供了额外的控制信息和功能扩展，可以反映出不同应用的网络行为特征。下面是一些常见的 TCP 选项及其在不同应用流量中的可能差异：

最大报文段长度（MSS）：这是最常见的 TCP 选项之一，它定义了 TCP 报文段的最大长度。对于需要高效率传输大量数据的应用（如文件传输、视频流），通常会设置较大的 MSS 值。相对地，对于交互式应用（如网页浏览、即时通信），可能会使用较小的 MSS 值以减少延迟。

窗口扩大因子（Window Scale）：用于扩大 TCP 窗口大小的范围，允许更大的窗口值，从而提高传输效率。在需要传输大量数据的应用中更常见，如视频流媒体或大规模数据传输，它们可能会利用窗口扩大因子来提升吞吐量。

选择性确认（SACK）：允许接收方指明哪些数据被成功接收，而不是仅依靠累积确认。对于高速网络或那些在不稳定网络环境下运行的应用（如移动网络下的数据传输），选择性确认可以提高效率和可靠性。

时间戳：用于计算往返时间（RTT）和避免序列号的回绕问题。时间戳在需要精确测量网络延迟或在高速网络中运行的应用中更为常见。

快速打开（TCP Fast Open）：允许数据在初始的 SYN 包中传输，减少了连接建立的往返时间。这种选项可能在需要快速建立连接的应用中使用，如网页浏览或某些 API 调用。

通过分析 TCP 选项字段的使用模式，可以推断出应用的类型和网络行为。例如，大量使用窗口扩大因子和大 MSS 值的流量可能来自于数据密集型应用，而频繁利用快速打开选项的流量可能表示轻量级或交互性强的网络活动（WeChat）。

5.7.6 IP 标识、MAC 类型

使用 IP 标识和 MAC 类型来区分不同的应用流量可能具有一定的挑战性，因为这些字段通常不直接与应用层数据相关联。不过，它们在某些情况下可以提供有用的间接信息：

IP 标识：在 IPv4 中，IP 标识用于标记属于同一数据报的各个分片。这个标识在每个发出的数据报中是唯一的。对于分片较多的流量，如视频或大文件传输，可能会观察到更频繁的 IP 标识变化。然而，由于现代网络通常避免使用 IP 分片，以及 IPv6 中对分片的处理方式不同，IP 标识的作用可能不如过去明显。

MAC 类型：MAC 类型（或以太网类型字段）标识了上层使用的协议类型，例如 IPv4、IPv6、ARP 等。通过观察 MAC 类型，可以区分使用不同网络协议的流量。例如，IPv6 流量可能与最新的应用或服务相关联，而 IPv4 流量可能更普遍。在某些特定的网络环境或企业网络中，特定的 MAC 类型可能与特定的应用或服务关联，但这种关联通常需要特定网络环境的先验知识。

总体来说，虽然 IP 标识和 MAC 类型可以提供一些关于网络流量性质的线索，但它们通常不足以直接区分不同的应用流量。要准确地识别和分类应用流量，还需要结合更多的数据，如传输层协议的特性、应用层数据的模式等。这种复合方法更可能提供准确的流量分类。

5.7.7 IP 生存时间

IP 生存时间（TTL）：表示数据包在网络中的生存时间。不同的应用可能会设置不同的 TTL 值，这反映了不同网络路径和应用特性。

5.7.8　IP 数据包总长度

IP 数据包总长度：数据包的长度可以提供关于传输数据量的信息。不同的应用（如文本通信和视频流）可能会生成具有显著不同长度的数据包。

5.7.9　IP 协议字段

IP 协议字段 proto：指明了上层使用的协议类型（如 TCP、UDP）。这对于区分基于不同协议的应用流量至关重要，用于指示封装在 IP 数据包内的上层协议类型。这个字段对于区分基于不同协议的应用流量至关重要，因为它直接告诉我们数据包是属于哪种类型的传输协议。以下是一些常见的 proto 值及其在流量分类中的作用：

TCP（传输控制协议）- proto 值为 6：TCP 是一种可靠的、面向连接的协议，广泛用于 Web 浏览、电子邮件、文件传输等应用。如果 IP 数据包的 proto 字段显示为 TCP，这通常意味着数据包是属于需要建立稳定连接的应用。

UDP（用户数据报协议）- proto 值为 17：UDP 是一种简单的、面向数据报的协议，常用于视频和音频流、在线游戏、某些类型的 VPN 等。UDP 流量的特点是低延迟，但不保证数据包的顺序或完整性。如果 proto 字段显示为 UDP，这通常表示数据包属于对实时性要求较高的应用。

ICMP（因特网控制消息协议）- proto 值为 1：ICMP 主要用于网络诊断和错误报告，如 ping 命令。如果数据包的 proto 字段是 ICMP，这通常表示这些包是用于网络维护或故障排查。

IGMP（因特网组管理协议）- proto 值为 2：IGMP 用于管理多播组成员身份。IGMP 流量表明网络上可能存在多播传输，这在某些类型的视频传输或实时数据流中比较常见。

通过检查 IP 数据包的 proto 字段，可以快速地对流量进行初步分类，区分出使用不同协议的应用类型。这对于网络管理、安全监控、以及优化网络性能等方面非常有用。然而，要深入理解流量的具体应用类型，还需要结合其他信息，如端口号、数据包内容、流量模式等。

5.7.10　IP 标志和片偏移

IP 标志和片偏移：这些用于处理 IP 数据包的分片和重组。不同应用产生的数据包可能有不同的分片和重组模式，有助于流量分类。

IP 版本和首部长度：这表明了 IP 数据包的协议版本（如 IPv4 或 IPv6）和首部长度。不同的应用可能偏好不同的 IP 版本，这可以作为区分它们的一个依据。0 量问题或具有重试机制的应用相关。通过分析 TCP 序列号和确认号的变化模式，可以对应用的通信行为进行一定程度的推断。然而，这种分析通常需要结合其他网络特征和上下文信息，才能更准确地识别特定应用。

5.7.11 IP 首部校验和

实际上，IP 首部校验和并不直接反映不同应用产生的流量类型。IP 首部校验和是一个网络层的功能，用于确保 IP 头部在传输过程中的完整性和正确性。它是一个计算出的值，用于检测数据在传输过程中是否发生了错误或变化。对于不同的应用来说，它们产生的流量的 IP 首部校验和值并不是一个区分应用类型的可靠指标。这是因为：标准化计算：IP 首部校验和是按照标准化的算法计算的，这个算法对于所有的 IP 数据包都是一样的。它的值完全取决于 IP 头部的内容（如源 IP 地址、目的 IP 地址、协议类型等），而这些内容与应用类型没有直接关联。传输层以上的信息不影响：IP 首部校验和仅仅检查和保护 IP 头部的信息，它不涉及传输层（如 TCP 或 UDP）或更高层次（如应用层数据）的信息。因此，不同应用的流量在 IP 层面上可能非常相似。

5.8 本章小结

本章节我们首先分析了使用深度学习可解释性算法实现数据轻量化的动机和必要性，针对现有的数据轻量化问题，我们提出了基于可解释性算法进行数据轻量化的研究。我们根据现有的数据集优化技术筛选出适合的优化技术。首先，我们采用上一章介绍的深度模型进行流量识别分类训练。然后，我们采用 SHAP 可解释性算法计算每个样本中各个特征的贡献值。并将所有样本的对应特征贡献度相加得到各个特征的总贡献度。实验表明相比于加密流量，未加密流量可以提取到更多可靠的特征，更加有助于流量识别分类。例如在资源有限的情况下我们可以使用贡献度高的特征字段进行流量识别。

第 6 章　基于知识蒸馏的模型轻量化方法

本章节，我们首先通过大模型以及工程实践的例子分析现有的网络流量识别技术模型复杂的问题；然后，我们根据第二章介绍的模型轻量化技术，详细阐述了知识蒸馏的工作原理以及相关概念，并阐述了知识蒸馏的工作过程；紧接着，我们展示了基于知识蒸馏的模型轻量化算法；最后，我们将提出的方法在测试集上进行了实验评估，分析了实验结果、并与相关的工作进行了比较和讨论。WiKibookBibliography

6.1　研究动机

在当今大型模型（如深度学习神经网络）流行的背景下，对于网络流量识别任务来说，使用简单轻量级模型依然具有重要性。轻量级模型的主要优点包括较低的计算需求和更快的推理速度，这在资源受限的环境（如边缘计算设备）中尤为重要。此外，轻量级模型通常更易于部署和维护，尤其是在动态变化的网络环境中。相比之下，大型模型通常提供更高的准确性和强大的泛化能力，但这种性能提升往往以较高的计算成本、更长的训练时间和更大的存储需求为代价。例如，大型模型如 BERT 或 GPT 系列在自然语言处理任务中表现出色，但它们需要显著的计算资源WiKibookBibliography。在工程实践中，选择模型时需要综合考虑准确性、计算资源、实时性需求和部署环境。轻量级模型适合于资源受限或对实时性要求较高的场景，而大型模型更适用于计算资源充足且对准确性要求较高的应用场景。因此，两者并不是相互替代的关系，而是根据具体需求和条件选择最合适的模型。

在第 2 章我们阐述了模型轻量化的几类技术：第一类技术模型剪枝（Model Pruning）是通过去除模型中冗余的连接或参数，减小模型的大小。这可以通过权重修剪、通道修剪或层修剪等方式实现。剪枝后的模型通常具有更小的体积和更快的推理速度。第二类量化是将模型参数从浮点数转换为较小的整数或低精度浮点数的过程。这降低了参数表示的位数，减小了模型的内存占用和计算需求。通常，8 位整数或更低位的量化方法被广泛使用。第三类轻量级网络设计，使用更轻量、结构简单的神经网络架构，如 MobileNet、SqueezeNet 等。这些网络结

构专注于在减小参数数量的同时保持较高的性能。第四类知识蒸馏（Knowledge Distillation）通过在训练过程中利用一个教师模型的知识来训练一个小而简单的学生模型。这样可以在保持性能的同时，减小模型的体积。知识蒸馏不仅可以轻量化模型，还有助于提高模型的泛化能力。我们根据现有的模型轻量化技术，结合流量识别外部环境，提出使用知识蒸馏技术进行模型轻量化。

6.2 知识蒸馏相关概念

WiKibookBibliography 知识蒸馏（Knowledge Distillation）是一种迁移学习的技术，旨在通过将一个复杂且性能强大的模型的知识传递给一个更为轻量的模型，从而在保持模型体积较小的同时提升性能。这种方法最初由 Hinton 等人于 2015 年提出，其核心思想是借助教师模型的“软标签”（soft labels）来引导学生模型的训练。在知识蒸馏中，教师模型通常是一个深层次的、准确率较高的模型，而学生模型则是一个相对简化的模型，旨在在资源有限的设备上执行。

知识蒸馏的损失函数通常结合了两个部分：交叉熵损失和 KL 散度损失。交叉熵损失确保了学生模型在训练中能够正确预测样本的类别，而 KL 散度损失则用于度量教师模型和学生模型输出概率分布之间的相似性，确保学生模型能够复制教师模型的决策边界和软标签信息。温度参数用于平衡这两个损失项，使得知识的传递更为平滑，有助于提高模型泛化性能。

知识蒸馏在实际应用中广泛用于解决两个主要问题：首先，它允许在计算资源受限的环境中部署轻量级模型，如嵌入式设备或移动端；其次，通过引入教师模型的知识，知识蒸馏有助于提升学生模型在特定任务上的性能，即使学生模型相对较小。这种技术已经在图像识别、自然语言处理等多个领域取得了显著的成功，并为实际应用中的模型设计提供了一种有效的迁移学习策略。

如上图所示，教师网络（左侧）的预测输出除以温度参数（Temperature）之后、再做 Softmax 计算，可以获得软化的概率分布（软目标或软标签），数值介于 0 – 1 0-11 之间，取值分布较为缓和。Temperature 数值越大，分布越缓和；而 Temperature 数值减小，容易放大错误分类的概率，引入不必要的噪声。针对较困难的分类或检测任务，Temperature 通常取 1，11，确保教师网络中正确预测的贡献。硬目标则是样本的真实标注，可以用 One-hot 矢量表示。Total loss 设计为软目标与硬目标所对应的交叉熵的加权平均（表示为 KD loss 与 CE loss），其中

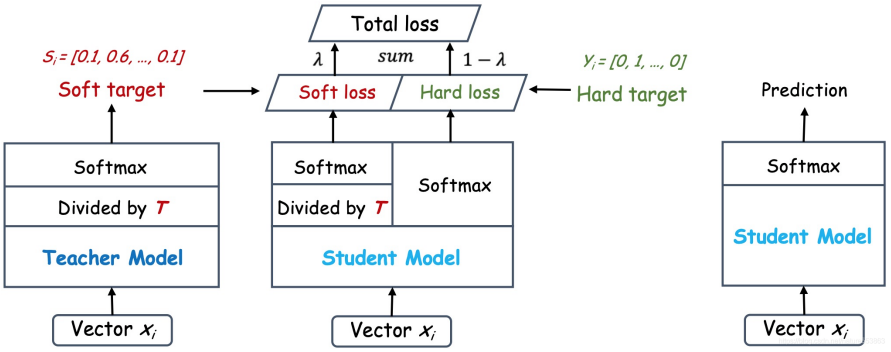

图 6.1 知识蒸馏过程

软目标交叉熵的加权系数越大，表明迁移诱导越依赖教师网络的贡献，这对训练初期阶段是很有必要的，有助于让学生网络更轻松的鉴别简单样本，但训练后期需要适当减小软目标的比重，让真实标注帮助鉴别困难样本。另外，教师网络的预测精度通常要优于学生网络，而模型容量则无具体限制，且教师网络推理精度越高，越有利于学生网络的学习。教师网络与学生网络也可以联合训练，此时教师网络的暗知识及学习方式都会影响学生网络的学习，具体如下（式中三项分别为教师网络 Softmax 输出的交叉熵 loss、学生网络 Softmax 输出的交叉熵 loss、以及教师网络数值输出与学生网络 Softmax 输出的交叉熵 loss）

6.3　算法设计

根据第五章可知，使用从传输层采集到的未加密应用流量在进行可解释性计算后可以得到一些可靠的特征进行精细化采集。并在实验结论中看到使用其输入到残差网络后的识别效果达到了不错的效果。故本次实验我们首先采用第四章所介绍搭建过的残差网络（Resnet50）作为知识蒸馏中的教师模型，使用传输层采集到的未加密流量作为模型的数据集。其次，我们定义了一个轻量化的模型（resnet18）作为学生模型，学生模型拥有比教师模型更少的参数。这样的设计使得学生模型更适合在资源受限的环境中进行部署，例如移动设备或边缘计算平台。

ResNet18 模型是一个深度残差网络，由 4 个不同层次的残差块组成，每个块中有 2 个基本的残差单元（BasicBlock）。每个 BasicBlock 由两个 3x3 卷积层组成，后接批量归一化和 ReLU 激活函数。如果需要下采样，会在块的首个 BasicBlock 中引入步长（stride）不为 1 的卷积，并通过额外的 1x1 卷积来调整维度。ResNet18

的特点是它的结构相对简单，参数数量适中，适合于计算资源有限的情况，而且它能够通过残差连接有效地训练较深的网络，减少梯度消失问题。ResNet50 与 ResNet18 在架构上的主要区别主要表现在残差块的设计、网络的层数、参数量以及计算复杂度上。ResNet50 采用了 Bottleneck 残差块，每个块由三个卷积层组成（分别是 1x1、3x3 和再次 1x1 卷积），这使得它能够在保持特征图尺寸不变的前提下，增加更多的特征通道。相比之下，ResNet18 使用的 BasicBlock 结构较为简单，仅包含两个 3x3 卷积层。正如模型名称所暗示的，ResNet50 由 50 层组成，而 ResNet18 仅有 18 层，因此 ResNet50 的参数数量远超 ResNet18。此外，ResNet50 的 Bottleneck 结构在特征通道上实现了四倍的扩展，而 ResNet18 的 BasicBlock 则没有这种扩展。由于层数和结构的差异，ResNet50 在捕捉更复杂特征的能力上优于 ResNet18，但相应地，它也需要更多的计算资源。因此，在实际应用中，选择哪种模型需综合考虑任务的复杂度和可用的计算资源。

为了在知识蒸馏中更有效地传递教师模型的知识，我们采用了两个关键概念的损失函数：硬标签损失（hard loss）和软标签损失（soft loss）。硬标签损失用于度量学生模型的预测与实际标签的一致性，这有助于确保学生模型在基本分类任务上表现出色。而软标签损失关注于教师模型输出与学生模型输出之间的距离，这使得学生模型能够更细致地学习到教师模型的决策边界和特征表示。在知识蒸馏的训练过程中，我们通过综合考虑硬标签损失和软标签损失，定义了总损失（Total Loss）。通过调整损失的权重超参数，我们能够平衡两者之间的影响，使得学生模型能够在相对较小的模型规模下，充分吸收教师模型的知识。这样的设计不仅在轻量化模型方面取得了显著效果，同时保留了模型的高性能，为实际应用提供了可行的解决方案。通过这一蒸馏过程，学生模型得以在保持较小体积的同时，具备了与教师模型相媲美的预测能力。

6.4 实验评估

我们对深度学习模型训练出来的效果在预留出的测试集上进行测试。我们所使用的是较新的 PyTorch v2.0.1 框架进行搭建；使用的 Docker 环境是 v20.10.10 版本，Python 采用的是 v3.10.12 版本，TensorFlow 采用的是 v2.13.0 版本。

6.4.1　度量指标

KFold 交叉验证是一种模型评估方法，它将数据集分割成 K 个子集。在 K 次的迭代中，每次选择一个子集作为测试集，而其余的 K-1 个子集作为训练集。这种方法有助于减少模型评估的偏差和方差，因为每个数据点都有机会作为测试数据使用。KFold 交叉验证特别适用于小数据集，能够更有效地利用有限的数据资源。

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Samples}} \tag{6.1}$$

训练损失是在训练过程中计算的，它表示模型在训练数据上的性能。通常，这是通过计算模型在训练集上每个样本的预测误差并求和得到的。测试损失则是在测试数据上计算的，它表示模型在未见数据上的性能。这两个指标都是评估模型拟合程度的重要工具。准确率（Accuracy）是最常用的性能指标之一，它简单地衡量模型预测正确的样本占总样本的比例。

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{6.2}$$

精确率（Precision）和召回率（Recall）则用于在二分类问题中更细致地评估模型性能。精确率是指模型正确预测为正类的样本数占模型预测为正类的总样本数的比例.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{6.3}$$

而召回率是指模型正确预测为正类的样本数占实际正类的总样本数的比例。在不平衡的数据集中，这两个指标尤其重要，因为它们可以帮助理解模型在预测不同类别时的性能。

6.4.2　实验结果

表 6.1 知识蒸馏教师学生网络参数表

Table 6.1 Knowledge distillation teacher student network parameter table

模型	网络类型	参数数量 (Millions)	残差块类型	层数
学生网络	ResNet18	11.18	BasicBlock	18
教师网络	ResNet50	25.56	Bottleneck	50

通过上述表格可以观察到，ResNet18 作为学生模型，拥有较少的参数数量（11.18 百万）以及相对简单的 BasicBlock 残差块结构，总层数为 18。而 ResNet50

作为教师模型，具有更多的参数数量（25.56 百万）和复杂的 Bottleneck 残差块结构，总层数为 50。知识蒸馏的优势在于通过训练学生模型，使其能够从教师模型中获取更为丰富的知识。在这个具体的比较中，ResNet50 作为教师模型拥有更深、更复杂的网络结构，能够捕捉更多抽象和复杂的特征，提供了更为丰富的知识。通过知识蒸馏，这些丰富的知识可以被传递给 ResNet18 作为学生模型，使得 ResNet18 在相对较少的参数和层数下，也能够具备对复杂特征的学习能力。这种迁移学习的方式有效地提高了 ResNet18 的性能，使其在资源受限的环境中也能够表现出色。此外，ResNet18 和 ResNet50 的残差块类型分别为 BasicBlock 和 Bottleneck，这也是知识蒸馏中的一项重要考虑因素。知识蒸馏并非简单地将所有知识直接复制到学生模型，而是通过软目标（soft targets）的方式，引导学生模型更好地学习复杂任务。因此，ResNet18 通过学习 ResNet50 的知识，不仅能够在参数和层数上实现轻量化，还能够通过教师模型更高级的特征学习，提高对抽象特征的理解和泛化能力。

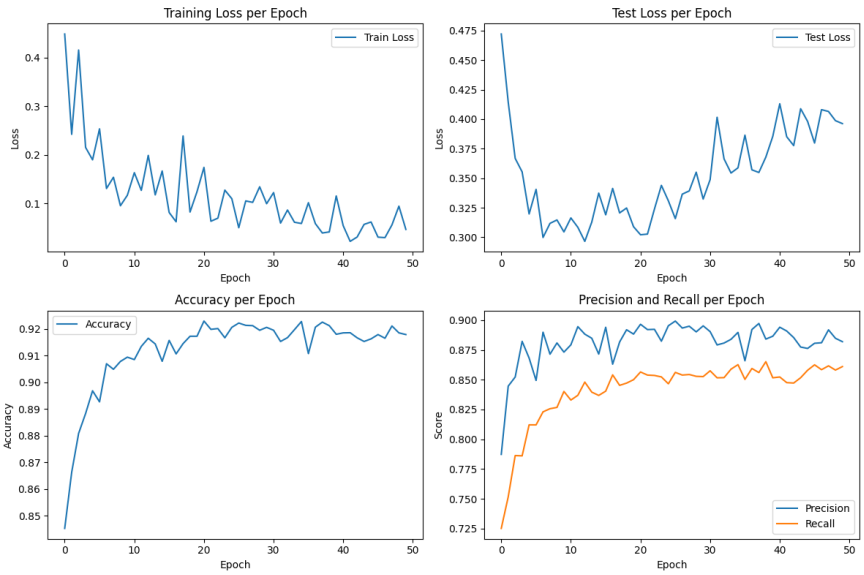

图 6.2 Resnet18 分类结果

在模型训练与测试的过程中，我们通过对损失、准确率、精确率和召回率的动态观察，进行了模型性能的深入分析。如图 6.2，从训练损失图可以看出，模型随着时间的推移学习能力增强，损失逐渐降低，但波动较大，表明可能学习率设置偏高或数据复杂性较高。测试损失图显示出波动上升趋势，暗示模型可能出现过拟合现象。准确率图表明模型在分类正确性上随着时间呈稳定提升。精确率与召回率的走势则显示模型在预测正类方面变得更为精确和全面。整体而

言，模型性能在准确度和精确率-召回率平衡方面有所提升，但需注意避免过拟合问题，可能需要通过调整正则化策略、学习率或提前终止训练来优化性能。从提供的模型性能图表中，我们可以得出以下结论：训练损失下降趋势：模型在训练过程中逐步学习，损失减少，但存在波动，可能需要调整学习率或考虑模型复杂性。测试损失上升：模型在测试集上的损失波动并趋于上升，这表明模型可能过拟合，对未知数据的泛化能力下降。准确率稳定：模型的准确率提升并趋于稳定，说明模型具有较好的识别能力。精确率与召回率：精确率和召回率随着训练的进行都有所提高，但精确率普遍高于召回率，表明模型在正类预测上较为谨慎。总体来说，模型在准确性、精确度和召回率方面表现良好，但是随着训练的进行，测试损失的上升提示需要注意过拟合问题，并考虑采取措施如正则化、调整学习率或早停等策略来优化模型性能。

6.5　讨论

知识蒸馏（Knowledge Distillation）作为一种模型轻量化技术，相较于其他方法具有独特的优点，为在资源受限或移动端环境中部署深度神经网络提供了一种有效的解决方案。首先，知识蒸馏通过在训练过程中将大型、复杂的教师模型的知识传递给小型、轻量的学生模型，实现了模型的压缩。这一过程中，采用了软目标（soft targets）的概念，使得学生模型可以更充分地学习到教师模型的知识，而非仅仅是其预测结果。这种软目标的使用不仅有助于提高学生模型在训练集上的性能，还能够使得学生模型更好地泛化到未见过的数据，从而在实际应用中表现更为鲁棒。其次，知识蒸馏能够在一定程度上缓解深度神经网络在轻量设备上的计算和存储资源需求。通过将教师模型的知识以更为紧凑的方式嵌入学生模型中，可以显著减少学生模型的参数量和模型大小，从而减小模型在推理阶段所需的计算资源，适应于嵌入式设备、移动端等资源受限环境。这对于实现实时性能和降低能耗具有重要意义，尤其是在物联网和边缘计算等领域。另外，知识蒸馏还具有更好的可解释性。传统的轻量化技术，如剪枝、量化等，往往会导致模型结构的破碎，难以保持原始模型的结构和语义信息。而知识蒸馏通过在学生模型中融入教师模型的知识，使得学生模型更具有可解释性。学生模型可以学习到教师模型对于输入数据的抽象表示和关键特征，有助于更清晰地理解模型在决策过程中的依据。这对于深度学习模型的可信度和可控性提供了更多的保

障，特别是在一些对解释性要求较高的应用场景，如医疗诊断和自动驾驶等。此外，知识蒸馏还能够在一定程度上提高模型的鲁棒性。通过在训练过程中引入教师模型的知识，学生模型更容易学习到对抗性样本的鲁棒特性，提高模型对于输入扰动的容忍度。这对于应对真实世界中的噪声和干扰，以及对抗攻击等具有积极意义。在网络安全领域，知识蒸馏可以为模型的抗攻击性能提供额外的支持，增强模型的安全性。

然而，需要注意的是，知识蒸馏并非没有挑战。首先，由于知识蒸馏依赖于教师模型的存在，其性能很大程度上受制于教师模型的质量和复杂度。如果教师模型过于简单或者过于复杂，可能会影响到知识的传递效果。其次，软目标的选择和调整对知识蒸馏的效果有很大的影响，需要在实践中仔细调优。最后，知识蒸馏对于一些复杂任务或者需要大量领域专业知识的任务可能效果有限，因为教师模型的知识可能无法充分覆盖所有的任务细节和领域知识。综合而言，知识蒸馏相比其他模型轻量化技术具有诸多优势，如更好的性能保持、更低的计算和存储需求、更好的可解释性和提高鲁棒性等。在资源受限环境中，特别是移动端和边缘设备上，知识蒸馏为部署深度学习模型提供了一种可行的途径，为实现轻量、高效的人工智能应用打开了新的可能性。

6.6 本章小结

本章节我们首先分析了使用深度学习进行流量识别的动机和必要性，针对现有的流量识别问题，我们提出了基于 Resnet50 网络进行流量识别研究。我们根据现有的深度学习流量识别方法筛选出适合进行特征贡献度计算的神经网络。首先，我们采用控制变量法探索了报头字段和数据字段在流量识别中的相对重要性。然后，我们从应用层处理开放式系统互联模型（OSI）中的不同层中采集到的数据应用在了搭建好的神经网络中。并通过准确率等指标对识别效果进行了评估展示。实验表明相比于数据字段，报头字段更加有助于流量识别分类。例如在资源有限的情况下我们可以有限处理报头字段。其次，实验表明使用 Resnet50 网络对于流量识别任务具有高准确率的优点。

第 7 章　总结与展望

7.1　总结

本论文面向流量识别中存在的模型数据轻量化展开研究，针对传统流量识别技术中存在的泛化性低的问题；针对流量识别技术在实际运行时面临的硬件资源有限，面对大量数据训练时间慢的问题，具体研究了轻量化流量识别的研究。接下来，我们从下面几个方面阐述本论文具体的贡献：

- 第一，针对节约实际工程环境中计算机存储资源和训练时间的问题，我们提出了基于可解释性算法的数据集优化方法，减少了深度学习模型对低贡献度特征的关注，从而达到了提高重要特征在所在数据集中的占比效果。与现有的方法进行了对比，我们的方法将现有数据集大小精简为原来的四十六分之一（1/46）。

- 第二，针对实际工程环境中模型参数复杂，模型训练时间长等问题，我们提出了基于知识蒸馏的模型优化方法，我们将之前做流量识别的复杂网络设为教师网络，在此基础上我们定义了一个轻量级的深度学习模型作为学生模型。该方法减少了模型卷积层数量，降低了模型时延并压缩了模型参数。与现有的方法对比，我们的方法缩短了模型的训练时间（从原来的 12 小时缩短为 2 小时）。模型参数由原来的约 25.56 百万缩减为约 11.18 百万，即缩减为原来的四分之一。

7.2　展望

诚然，本论文还有一些需要继续深入探索的科学和工程问题：

- 第一，第五章使用的 SHAP 可解释性算法，SHAP 值的解释依赖于底层模型的特性，对于一些复杂的非线性模型，SHAP 值的解释性可能受到限制。值得注意的是 shap 模型适用于多种学习模型，本论文是阅读参考了 SHAP 技术的贡献度计算方法并将其运用到残差网络中，这属于一个工程问题，并不牵扯到理论的创新，所以此工程任务留作未来的工作。

- 第二，第四章使用的基于深度学习的流量识别方法，其模型选择考虑到后续使用可解释性算法能够计算出重要特征（具有物理意义）从而选择了善于进行图形处理的残差网络。值得注意的是使用流量识别还是很多合适的模型，这也是

一个工程问题，我们将对其其他模式的识别方式这个问题留作以后的工作。

- 第三，由于本实验在做工程测试时采集处理的都是未加密的流量，我们实验进行流量识别实验以及可解释性计算时重点偏向于未加密流量。对于加密流量的识别处理以及涉及到的一些工程问题，需要进一步深入的研究。

附录 A　中国科学院大学学位论文撰写要求

学位论文是研究生科研工作成果的集中体现，是评判学位申请者学术水平、授予其学位的主要依据，是科研领域重要的文献资料。根据《科学技术报告、学位论文和学术论文的编写格式》（GB/T 7713-1987）、《学位论文编写规则》（GB/T 7713.1-2006）和《文后参考文献著录规则》（GB7714—87）等国家有关标准，结合中国科学院大学（以下简称“国科大”）的实际情况，特制订本规定。

表 A.1 这是网络特征表。

Table A.1 This is a table of network features.

分类	特征名称	字段大小
MAC 数据报报头	MAC 目的地址	1
	MAC 源地址	2
	MAC 类型	3
IP 数据报报头	版本首部长度	5
	区分服务	6
	总长度	9
	IP 标识	9
	IP 标志片偏移	9
	生存时间	9
	协议	9
	首部校验和	9
	IP 源地址	9
	IP 目的地址	9
TCP 数据报报头	16 位源端口号	5
	16 位目的端口号	9
	32 位序列号	9
	32 位确认序号	9
	首部长度保留位标志位	9
	16 位窗口大小	9
	选项字段	9

A.1 论文无附录者无需附录部分

A.2 测试公式编号 $\Lambda, \lambda, \theta, \bar{\Lambda}, \sqrt{S_{NN}}$

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{V}) = 0 \\ \frac{\partial (\rho \boldsymbol{V})}{\partial t} + \nabla \cdot (\rho \boldsymbol{V}\boldsymbol{V}) = \nabla \cdot \boldsymbol{\sigma} \\ \frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho E \boldsymbol{V}) = \nabla \cdot (k \nabla T) + \nabla \cdot (\boldsymbol{\sigma} \cdot \boldsymbol{V}) \end{cases} \quad \ldots \text{(A.1)}$$

$$\frac{\partial}{\partial t} \int_{\Omega} u \, \mathrm{d}\Omega + \int_{S} \mathbf{n} \cdot (u \boldsymbol{V}) \, \mathrm{d}S = \dot{\phi} \quad \ldots \text{(A.2)}$$

$$\mathcal{L}\{f\}(s) = \int_{0^-}^{\infty} f(t) e^{-st} \, \mathrm{d}t, \ \mathscr{L}\{f\}(s) = \int_{0^-}^{\infty} f(t) e^{-st} \, \mathrm{d}t$$

$$\mathcal{F}\big(f(x + x_0)\big) = \mathcal{F}\big(f(x)\big) e^{2\pi i \xi x_0}, \ \mathscr{F}\big(f(x + x_0)\big) = \mathscr{F}\big(f(x)\big) e^{2\pi i \xi x_0}$$

mathtext: $A, F, L, 2, 3, 5, \sigma$, mathnormal: $A, F, L, 2, 3, 5, \sigma$, mathrm: $\mathrm{A, F, L, 2, 3, 5, \sigma}$.

mathbf: $\mathbf{A, F, L, 2, 3, 5, \sigma}$, mathit: $\mathit{A, F, L, 2, 3, 5, \sigma}$, mathsf: $\mathsf{A, F, L, 2, 3, 5, \sigma}$.

mathtt: $\mathtt{A, F, L, 2, 3, 5, \sigma}$, mathfrak: $\mathfrak{A, F, L, 2, 3, 5, \sigma}$, mathbb: $\mathbb{A, F, L, 2, 3, 5, \sigma}$.

mathcal: $\mathcal{A, F, L, 2, 3, 5, \sigma}$, mathscr: $\mathscr{A, F, L, 2, 3, 5, \sigma}$, boldsymbol: $\boldsymbol{A, F, L, 2, 3, 5, \sigma}$.

vector: $\boldsymbol{\sigma}, \boldsymbol{T}, \boldsymbol{a}, \boldsymbol{F}, \boldsymbol{n}$, unitvector: $\boldsymbol{\sigma}, \mathbf{T}, \mathbf{a}, \mathbf{F}, \mathbf{n}$

matrix: $\boldsymbol{\sigma}, \mathbf{T}, \mathbf{a}, \mathbf{F}, \mathbf{n}$, unitmatrix: $\boldsymbol{\sigma}, \mathbf{T}, \mathbf{a}, \mathbf{F}, \mathbf{n}$

tensor: $\boldsymbol{\sigma}, \mathbf{T}, \mathbf{a}, \mathbf{F}, \mathbf{n}$, unittensor: $\boldsymbol{\sigma}, \mathbf{T}, \mathbf{a}, \mathbf{F}, \mathbf{n}$

A.3 测试生僻字

参考文献

陈浩元. 著录文后参考文献的规则及注意事项 [J]. 编辑学报, 2005, 17(6): 413-415.

陈晋镳, 张惠民, 朱士兴, 等. 蓟县震旦亚界研究 [M]//中国地质科学院天津地质矿产研究所. 中国震旦亚界. 天津: 天津科学技术出版社, 1980: 56-114.

初景利. 图书馆数字参考咨询服务研究 [M]. 北京: 北京图书馆出版社, 2004.

哈里森 · 沃尔德伦. 经济数学与金融数学 [M]. 谢远涛, 译. 北京: 中国人民大学出版社, 2012: 235-236.

牛志明, 斯温兰德, 雷光春. 综合湿地管理国际研讨会论文集 [C]. 北京: 海洋出版社, 2013.

袁训来, 陈哲, 肖书海. 蓝田生物群: 一个认识多细胞生物起源和早期演化的新窗口 – 篇一 [J]. 科学通报, 2012, 57(34): 3219.

袁训来, 陈哲, 肖书海. 蓝田生物群: 一个认识多细胞生物起源和早期演化的新窗口 – 篇二 [J]. 科学通报, 2012, 57(34): 3219.

袁训来, 陈哲, 肖书海. 蓝田生物群: 一个认识多细胞生物起源和早期演化的新窗口 – 篇三 [J]. 科学通报, 2012, 57(34): 3219.

ボハンデ. 過去及び現在に於ける英国と会 [J]. 日本時報, 1928, 17: 5-9.

Betts L R, Taylor C P. Aging reduces center-surround antagonism in visual motion processing [J]. Neuron, 2005, 45(3): 361-366.

Bravo H, Olavarria J. Comparative study of visual inter and intrahemispheric cortico-cortical connections in five native chilean rodents [J]. Anatomy and embryology, 1990, 181(1): 67-73.

Lamport L. Document preparation system [M]. Addison-Wesley Reading, MA, 1986.

Stamerjohanns H, Ginev D, David C, et al. MathML-aware article conversion from LaTeX [J]. Towards a Digital Mathematics Library, 2009, 16(2): 109-120.

Walls S C, Barichivich W J, Brown M E. Drought, deluge and declines: the impact of precipitation extremes on amphibians in a changing climate [J/OL]. Biology, 2013, 2(1): 399-418[2013-11-04]. http://www.mdpi.com/2079-7737/2/1/399. DOI: 10.3390/biology2010399.

Wikibook. http://en.wikibooks.org/wiki/latex [M]. On-line Resources, 2014.

ДубровинА. И. Открытое письмо Председателя Главного Совета Союза Русского Народа Санкт-Петербургскому Антонию, Первенствующему члену Священного Синода [J]. Вече, 1906: 1-3.

致 谢

感激 casthesis 作者吴凌云学长，gbt7714-bibtex-style 开发者 zepinglee，和 ctex 众多开发者们。若没有他们的辛勤付出和非凡工作，LaTeX 菜鸟的我是无法完成此国科大学位论文 LaTeX 模板 ucasthesis 的。在 LaTeX 中的一点一滴的成长源于开源社区的众多优秀资料和教程，在此对所有 LaTeX 社区的贡献者表示感谢！

ucasthesis 国科大学位论文 LaTeX 模板的最终成型离不开以霍明虹老师和丁云云老师为代表的国科大学位办公室老师们制定的官方指导文件和众多 ucasthesis 用户的热心测试和耐心反馈，在此对他们的认真付出表示感谢。特别对国科大的赵永明同学的众多有效反馈意见和建议表示感谢，对国科大本科部的陆晴老师和本科部学位办的丁云云老师的细致审核和建议表示感谢。谢谢大家的共同努力和支持，让 ucasthesis 为国科大学子使用 LaTeX 撰写学位论文提供便利和高效这一目标成为可能。

作者简历及攻读学位期间发表的学术论文与研究成果

作者简历:

casthesis **作者**

赵浩宇，江苏省徐州人，中国科学院信息工程研究所工程硕士研究生。

ucasthesis **作者**

莫晃锐，湖南省湘潭县人，中国科学院力学研究所硕士研究生。

已发表（或正式接受）的学术论文:

1. ucasthesis: A LaTeX Thesis Template for the University of Chinese Academy of Sciences, 2014.

申请或已获得的专利:

（无专利时此项不必列出）

参加的研究项目及获奖情况:

可以随意添加新的条目或是结构。