

# CycleLLH: 一种基于周期性整合的新型网络流量预测模型

唐文杰<sup>1)</sup> 肖一磊<sup>1)</sup> 孔祥宇<sup>1)</sup> 齐恒<sup>1)</sup> 刘秀龙<sup>2)</sup> 李克秋<sup>2)</sup>

<sup>1)</sup>(大连理工大学计算机科学与技术学院 辽宁 大连 116024)

<sup>2)</sup>(天津大学智能与计算学部 天津 300350)

**摘 要** 精准的网络流量预测是实现网络精细化和智能化管理的关键,也是网络运营商、云服务提供商等实现网络智能运维及应用服务保障的重要支撑,属于当前业界研究的热点.网络流量预测问题一般可被视为一种时间序列预测问题,现有时间序列预测模型虽然能起到一定作用,但这些通用模型很少考虑流量数据集本身特点,从而无法在网络流量预测性能上取得突破.为此,本文重点研究了网络流量数据集中的自然周期特征,提出了一种能有效利用时间序列周期性特点的网络流量预测通用模型——Cycle Little Linear Head(CycleLLH).该模型主干为Transformer的编码器,其中两个关键设计在于:(1)周期整合.将流量序列按照一个特定周期划分步长划分为不同的周期块,然后将这些周期块对应相位的时间节点分别嵌入到不同输入令牌;(2)小线性层.由多个多层感知机组组成,并且多层感知机单独作用于每个相位的时间特征.周期整合使得模型具有两个优点:更有利于模型提取数据集在一个周期内的特征;注意力矩阵的计算和内存复杂度可以看作是和周期划分步长二次方有关的常数,使得模型可以使用更大的回溯窗口而仅增加少量计算资源.通过在公共流量数据集上进行大量实验,本研究表明:与当前最先进的模型相比,CycleLLH在流量预测精度方面表现出显著优势,在六个数据集上的预测精度分别提升了12.3%、8.4%、29.9%、5.8%、8.3%和2.0%.代码可从<https://github.com/wenjetang218/CycleLLH.git>中获取.

**关键词** 网络流量预测;时间序列预测;周期性;周期整合;小线性层

中图法分类号 TP18

DOI号 10.11897/SP.J.1016.2024.02867

## CycleLLH: A New Network Traffic Prediction Model based on Cycle Integration

TANG Wen-Jie<sup>1)</sup> XIAO Yi-Lei<sup>1)</sup> KONG Xiang-Yu<sup>1)</sup> QI Heng<sup>1)</sup>

LIU Xiu-Long<sup>2)</sup> LI Ke-Qiu<sup>2)</sup>

<sup>1)</sup>(School of Computer Science and Technology, Dalian University of Technology, Dalian, Liaoning 116024)

<sup>2)</sup>(College of Intelligence and Computing, Tianjin University, Tianjin 300350)

**Abstract** Accurate network traffic prediction is not only a key link to realize network refinement and intelligent management, but also an important support for network operators and cloud service providers in intelligent operation and maintenance and application service guarantee. This field has become a research hotspot in the current industry, and has a wide range of application prospects. Network traffic prediction technology plays an important role in many fields, including base station power management, base station overload prevention, unmanned aerial vehicle temporary base stations, new network deployment and base station construction, 5G network slicing,

收稿日期:2024-01-15;在线发布日期:2024-09-13. 本课题得到国家自然科学基金面上项目(No. 62072069)资助. 唐文杰,硕士研究生,主要研究领域为时间序列预测. E-mail: 2458488571@mail.dlut.edu.cn. 肖一磊,博士研究生,主要研究领域为时间序列预测. 孔祥宇,博士研究生,主要研究领域为网络态势感知. 齐恒(通信作者),博士,教授,中国计算机学会(CCF)高级会员,主要研究领域为新一代互联网体系、数据中心网络. E-mail: hengqi@dlut.edu.cn. 刘秀龙,博士,教授,主要研究领域为物联网、区块链. 李克秋,博士,教授,国家杰出青年科学基金获得者、IEEE Fellow,主要研究领域为无线网络、云计算、软件定义数据中心网络.

Software Defined Network resource scheduling, and resource management in multi-access edge computing. Network traffic prediction problem is usually regarded as a time series prediction problem. With its self-attention mechanism, Transformer can efficiently compute the dependencies between each time node in parallel, and performs well in the field of time series prediction. However, although the existing time series prediction models, including the Transformer-based models, can play a certain role, they often ignore the unique periodic characteristics of traffic data sets, so it is difficult to make a breakthrough in network traffic prediction performance. Therefore, this paper focuses on the natural cycle of network traffic data sets, and proposes a general network traffic prediction model, Cycle Little Linear Head (CycleLLH), which can effectively exploit the periodicity of time series. The backbone of the model is the encoder of the Transformer, and the two key designs are as follows: 1. Cycle Integration: we divide the traffic sequence into different cycle blocks according to a specific period, and then embed the time nodes corresponding to the phase of these cycle blocks into different input tokens; 2. Little Linear Head: it is composed of multiple multi-layer perceptrons, and each multi-layer perceptron operates separately on each feature node. Cycle Integration makes the model have two advantages: it is more conducive to the model to extract the features of the dataset within one cycle; the computation and memory complexity of the attention matrix can be viewed as a constant related to the quadratic power of the period step of the dataset, allowing the model to use a larger look-back window with only a small increase in computational resources. The little linear layer preserves the spatial structure of the encoder output and significantly reduces the number of parameters in the linear layer, which helps the model to predict the detailed changes in the network traffic data. Through a large number of experiments on public traffic data sets, this study shows that CycleLLH shows significant advantages in traffic prediction accuracy compared with the current state-of-the-art models, and the prediction accuracy on the six data sets is improved by 12.3%, 8.4%, 29.9%, 5.8%, 8.3% and 2.0%, respectively. In addition, through comparative ablation experiments, this study verifies the effectiveness of Cycle Integration, Little Linear Head and normalization methods in improving the prediction performance of the model. Experimental results show that CycleLLH can achieve better prediction performance under longer backtracking window, and shows strong robustness and adaptability under different cycle division step sizes. In addition, CycleLLH also shows high stability in noisy data environment. Code can be obtained from <https://github.com/wenjetang218/CycleLLH>. git.

**Keywords** network traffic prediction; time series prediction; periodicity; cycle integration; little linear head

## 1 引 言

当前,云计算中心及云服务已成为新一代信息基础设施的重要形态.然而,国内外互联网企业,如微软、IBM、阿里、华为等,其云服务均屡现宕机事件,造成极大损失.云服务宕机的其中一个原因就是流量过载.网络流量的增加可能会对云服务的稳定性和可用性产生影响,从而增加了发生宕机的风

险<sup>[1-2]</sup>.为实现可靠的云服务质量保障,服务提供商应对用户访问量及应用流量进行精准地估计和预测,从而能及时预留资源或快速有效地提前分配资源,避免因资源不足所产生的服务宕机.为此,研究并构建高效可靠的网络流量预测模型尤为重要.

关于网络流量预测的应用研究目前受到业界高度重视<sup>[3]</sup>.通过精确的流量预测,服务提供商可为基站设计休息策略实现降低功耗<sup>[4]</sup>,当流量需求减少时,让基站停止工作或者处于低功耗模式而不影响

用户的使用. 随着网络流量需求增大, 网络繁忙的时段必然也会增加, 此时可以通过预测基站是否会达到过载, 主动接纳控制策略<sup>[5]</sup>, 允许或阻止特定应用、用户访问网络资源, 而不是仅在资源已经耗尽后才采取行动. 除了限制用户的连接, 现已有研究人员提出利用无人机作为临时基站补充地面流量系统来提高网络服务质量, 尤其是当自然灾害等意外情况发生导致地面基站无法工作时, 此时无人机的临时基站功能将更为重要<sup>[6]</sup>. 随着网络设备剧增, 增加建设基站势必会成为未来的一项大工程. 部署新网络时, 可以预测该地区高峰时段的流量, 来确定满足服务协议的最小容量<sup>[7]</sup>.

除此之外, 网络流量预测也将会为各种新技术的发展起到推动作用. 例如, 在5G广播技术中网络切片技术十分重要, 不同的切片是为不同的应用设备而建立的. 使用预测技术可以减少资源分配开销<sup>[8]</sup>, 扩大广播资源利用率, 保证网络用户的服务等级协议(service level agreement). 软件定义网络(software defined network, SDN)可以分离网络的控制和转发行为, 从而减少对硬件设备的依赖和对网络的可操作性, 使网络变得敏捷和灵活. SDN改进网络控制的方式是, 使服务提供商和企业快速响应不断变化的业务需求. 所以在SDN技术中, 可以设计基于预测的资源调度策略. SDN中的不同应用都需要用到网络流量预测技术, 例如虚拟网络功能布局与资源扩展、动态路由. 流量预测技术同样可用于促进多接入边缘计算(Multi-access Edge Computing, MEC)的部署和资源管理. 例如, Huang和Chen<sup>[9]</sup>考虑了智能城市中的移动流量卸载, 其中预测了未来一小时的最大、平均和最小流量需求, 并在MEC环境中设计了相应的卸载策略.

网络流量预测领域主要使用的模型有统计学模型、机器学习模型和深度学习模型. 随着人工智能的快速发展, 深度学习模型因其高性能、高准确率在预测中使用的占比越来越大, 近年来基于Transformer<sup>[10]</sup>的模型受到广泛关注. 利用Transformer解决时间序列预测问题存在大量相关研究, 每年都会有准确率更高、训练时间复杂度更小的模型出现. 网络流量预测的基本问题就是时间序列预测问题, 直接将这些基于Transformer的模型迁移到网络流量预测领域也是一种解决方法, 注意力机制和Transformer网络在网络流量预测问题中也被证明是有效的<sup>[11-13]</sup>. 但是这些模型毕竟都是时间序列预测领域的通用模型, 若能在模型设计时紧密结合网络流量

数据集独特的数据特性, 并在预测中充分利用, 有望构建出性能更卓越的模型.

综上, 本文旨在提出一种利用单变量网络流量数据特性的预测模型. 首先, 本文针对多个单变量网络流量数据进行大量研究后发现, 流量数据整体呈现出很强的周期性, 周期多为一周或者一天. 图1为一个典型网络流量数据集两周时间的图像, 不难看出: 以周为单位, 图像变化周期是七天, 前五天高峰, 后两天低峰; 以天为单位, 图像的走势展现出诸多相似之处, 比如周六周天的图像均为多个突起形状.

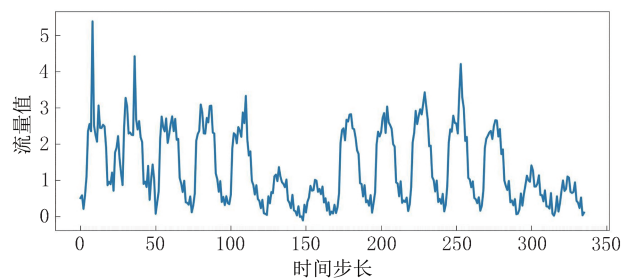


图1 网络流量数据两周的图像

基于这些发现, 本文提出了一种面向数据周期性的基于Transformer的模型CycleLLH. CycleLLH使用Transformer中的编码器作为主体架构, 将输入序列按照周期划分并整合后再作为编码器的输入令牌, 使得注意力机制能够有效利用网络流量数据中的周期性, 充分提取数据一个周期内的特征. 该方法将注意力矩阵的计算复杂度和周期划分步长 $C$ 联系起来而不是输入序列长度 $L$ , 大大降低了复杂度, 使得模型可以使用更长的回溯窗口.

除此之外, 对以往模型的预测图像进行分析, 发现大多数模型能预测网络流量数据的走势, 但是预测图像比较圆滑, 对数据的细节变化呈现不到位. 本文认为导致该问题的原因之一是以往模型使用的线性头(Linear Head, Transformer模块的输出和预测序列之间的神经网络层)是一个全连接层. 这会导致两个问题: 首先, 全连接层的设计可能会完全放弃时间序列特征的顺序结构; 其次, 大量的参数很容易导致过拟合.

于是, 本文赋予Transformer模块输出具体的含义, 经过周期整合后, 通过编码器能够精确提取到网络流量序列一个周期内的特征张量. 这个张量具有序列长度和特征个数两个维度, 张量的每一列代表提取的一个周期对应相位的时间特征. 进而提出小线性层的概念, 本文在每个相位的时间特征上, 分别



使用多层感知机 (MLP, Multilayer Perceptron), 将编码器输出的特征维度映射至一维, 和所需的预测序列构建出一种周期内相位对应的关系. 小线性层中多层感知机个数与输出序列的周期数保持一致.

本文的主要贡献如下:

(1) 本文重点研究了网络流量数据集中的周期特性, 将输入序列按照周期划分并整合后再作为编码器的输入, 使得模型能利用注意力机制充分提取一个周期内的特征.

(2) 为了改进以往模型无法预测到网络流量数据中的细节变化, 本文提出了小线性层的概念, 将编码器输出的特征序列和预测序列在周期内一一对应地联系起来.

(3) CycleLLH 在六个数据集上分别实现了 12.3%、8.4%、29.9%、5.8%、8.3% 和 2.0% 的预测精度提升, 其中周期划分步长分别为 168、48 和 24, 涵盖了周期为一周和一天的网络流量数据集.

论文剩余部分的结构安排如下: 第 2 节介绍本文模型涉及的 Transformer 神经网络、网络流量预测模型、周期整合和小线性层的相关工作, 第 3 节详细阐述本文模型的结构和优点, 第 4 节是实验结果与分析, 最后第 5 节进行结论总结并对未来工作进行展望.

## 2 相关工作

### 2.1 长时间序列预测和网络流量预测

自 Transformer 模型问世以来, 有大量工作研究如何将其应用于长时间序列预测. LogTrans<sup>[14]</sup> 使用卷积自注意力和 LogSparse 设计来捕获局部信息并降低空间复杂度. Reformer<sup>[15]</sup> 提出了基于局部敏感性哈希算法的注意力机制. Informer<sup>[16]</sup> 提出了一种 ProbSparse 自注意力提取机制, 以有效地提取最重要的特征. Autoformer<sup>[17]</sup> 借鉴了传统时间序列分析方法中的分解和自相关思想, 赋予了深度预测模型渐进分解的内在能力. FEDformer<sup>[18]</sup> 使用傅立叶增强结构来获得线性复杂度. Pyraformer<sup>[19]</sup> 应用了具有尺度间和尺度内连接的金字塔注意力模块, 也获得了线性复杂度. 这些模型基本上都侧重于设计新的机制以降低原始注意力机制的复杂度, 从而打破信息利用瓶颈, 最终在长时间序列预测中获得更好的预测性能.

在网络流量预测中, 深度学习模型经历了一系列的变革. 前馈神经网络 (Feed-Forward Neural

Network, FFNN) 是用于网络流量预测<sup>[20-21]</sup> 的最基本的深度神经网络. 随后卷积神经网络 (Convolutional Neural Networks, CNN) 和递归神经网络 (Recurrent Neural Network, RNN) 在网络流量预测领域<sup>[22-23]</sup> 开始崭露头角, 但它们依然有着梯度爆炸和消失等问题. 长短期记忆网络 (Long Short-Term Memory, LSTM) 应运而生, 并为网络流量预测<sup>[24-25]</sup> 带来了新的突破. 近年来, 不仅有研究者提出了 ConvLSTM (Convolutional LSTM) 这一创新模型, 还有研究者将长短期记忆网络与注意力机制相结合<sup>[26-27]</sup>, 进一步优化了预测效果, 为解决网络流量预测问题提供了新的思路. 但是要想在网络流量预测中成功运用注意力机制, 探究网络流量数据的特点至关重要.

### 2.2 时间序列预测中对输入序列的处理

在长时间序列预测领域, Transformer 模型已展现出令人瞩目的潜力. 尽管多数基于 Transformer 的模型仍沿用编码器-译码器的经典架构, 并通过优化原始注意力机制的复杂度以提升长序列训练效率及延长预测时间, 但是仍有部分研究聚焦于输入序列处理的深入探索.

PatchTST<sup>[28]</sup> 的主体架构只有 Transformer 中的编码器, 由于时间序列预测旨在了解每个不同时间步长的数据之间的相关性, 然而单个时间步长不像句子中的单词那样具有语义, 因此提出了 Patching 来提高模型提取局部语义信息的能力从而分析它们之间的连接. Patching 是将每个输入的单变量时间序列划分为可以重叠或不重叠的块然后再作为编码器的输入, 通过使用 Patching 可以大大减少输入令牌的数量, 使注意力图的内存使用和计算复杂度降低. 除此之外 TimesNet<sup>[29]</sup> 利用快速傅里叶变换 (FFT) 学习多个自适应周期, 将一维时间序列转换为基于这些周期的一组二维张量. 这种变换可以将周期内和周期期间的变化分别嵌入到二维张量的列和行中. 受 TimesNet 启发, WITRAN<sup>[30]</sup> 分析了时间序列的自然周期, 然后根据自然周期将序列从一维重新排列为二维, 以便模型分析长短期重复模式.

获此启发, 本文将输入的单变量网络流量时间序列按照一个周期将其划分为多个周期块, 然后将不同周期块对应相位的时间结点整合后作为一个输入令牌, 最后输入令牌的个数等于周期划分步长的大小, 使得注意力机制能更有效地提取数据一个周期内的特征.

### 2.3 计算机视觉中表征的处理

2009 年发布的 ImageNet 数据集为研究者们提

供了丰富的视觉数据,帮助他们在视觉数据的表征上实现了突破.同时,ImageNet挑战赛的发起,极大地推动了计算机视觉和机器学习研究的发展.在比赛中,众多基于LeNet<sup>[31]</sup>的优秀卷积神经网络架构崭露头角,包括AlexNet<sup>[32]</sup>、VGG<sup>[33]</sup>、NiN<sup>[34]</sup>和ResNet<sup>[35]</sup>等.

LeNet、AlexNet和VGG都使用全连接层处理卷积层和汇聚层提取的空间结构特征.NiN对这种设计模式提出质疑,认为使用全连接层可能会完全放弃先前学习到的表征的空间结构,进而提出在每个像素的通道上分别使用多层感知机的解决方案.AlexNet的网络参数大小高达230M,而NiN仅需29M,但NiN的性能却超过了AlexNet.这种在每个像素位置独立作用的多层感知机设计,极大地减少了网络参数量,有效避免了过拟合,展现出了极高的效率和性能.

获此启发,既然输入Transformer的张量是通过整合在周期上对应的时间结点得到的,不妨将其输出张量也看作是一个周期的特征序列,每个特征结点相对独立,并与一个周期的预测序列形成一一对应的关系.本文借鉴了NiN的思路,让多层感知机独立作用于每个特征结点,如同NiN在每个像素的通道上单独使用多层感知机一样.如图2所示,这样会大大减少参数量,并且充分利用Transformer模块输出的特征序列的顺序结构.这就是小线性层的核心设计思想.

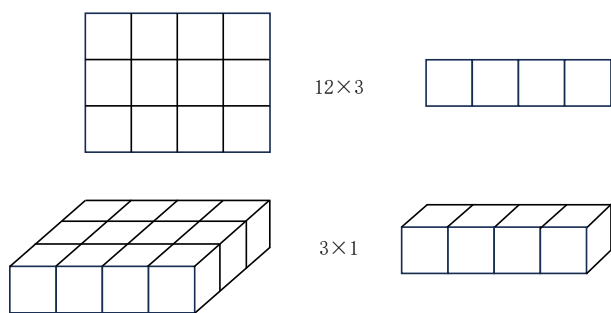


图2 将特征序列图像看作1维图像

### 3 模型架构

#### 3.1 模型简介

本文通过提出一个充分利用数据集周期特性的“Transformer”模型——CycleLLH来解决这个问题,该模型包含四个主要模块:

(1)周期整合. PatchTST通过将时间步长聚合

到子序列级别的补丁中,增强了局部性并捕获了点级别无法捕捉的全面语义信息.但是对于网络流量预测,更应该考虑其周期变化的数据特点,例如不同周期上对应的相位之间的相互联系.将输入序列中不同周期相同相位的时间结点看作一组组令牌进行输入,可以使编码器更好捕获一个周期内的变化规律,并按照规律精准影响被预测的结点.对于单变量网络流量,周期整合后的输入张量包含序列长度和特征个数两个维度.

(2)数据倍增.对于按周期划分并整合好后的一个周期的序列,不是直接输入到编码器中,而是将其复制 $w_1$ 份,然后在特征个数的维度上拼接再输入到编码器中.比如原序列为 $x \in R^{2 \times C}$ ( $C$ 代表周期划分步长),复制并拼接后的序列为 $x \in R^{(2 \times w_1) \times C}$ .设计的初衷在于,通过增加输入编码器中张量的特征个数维度,以促进编码器对特征的深入学习.在有限的回溯窗口中,数据集的周期数量是受到限制的,这意味着在完成周期整合后,输入张量的特征个数维度将会变得相对较小.在编码器中,主要的可学习参数是围绕特征个数维度进行线性变换的,这不利于编码器进行有效的特征学习.然而,通过数据倍增,我们可以增加特征数量的维度,从而更好地训练编码器进行特征学习.

(3)小线性层.小线性层由多个多层感知机组成.MLP的数量与所需预测的输出序列周期数量保持一致,每个MLP充当单周期预测器的角色,旨在生成特定周期的预测序列.与以往方法不同,本架构不再直接将编码器的输出铺平后通过线性层映射到 $T$ 维空间(其中 $M$ 为展开后的序列长度, $T$ 为被预测的序列长度).相反,我们将编码器的输出作为输入,通过多个MLP将特征维度映射到一维空间,以生成多个独立的周期预测序列.这些预测序列随后被拼接,并截取特定长度的部分,最终形成长度为 $T$ 的预测序列.此方法不仅提高了预测的精确性,而且通过引入多个MLP作为单周期预测器,增强了模型对复杂周期性模式的捕捉能力,使得生成的预测序列更加准确和可靠.

(4)稠密神经网络.在模型架构设计之初,我们构建了一个仅包含周期整合和小线性层的模型.经过实验验证,该模型也具有不错的预测效果.我们推断,当特征张量与预测序列在周期上对应相位形成一一映射时,原始序列对于预测序列的精确性具有重要影响.基于这一认识,我们采用了稠密神经

网络的架构,将未经神经网络模块处理的原始序列特征与编码器的输出在特征个数的维度上拼接,然后通过小线性层共同映射到一维空间.经实验验证稠密神经网络有助于提高模型的预测性能,具体实验结果请参考附录D.

本文设计的模型架构有以下优点:

(1)更好地利用了数据集的周期性.输入数据以一个特定周期为单位被划分为多个周期块,将这些周期块按照对应相位整合后得到输入至编码器的令牌组,这样的令牌组可以看作是富含一个周期内各个相位之间联系的张量.这样处理后的输入数据更有助于编码器学习到一个周期内的时间节点的变化规律.而小线性层使得每个被预测的结点都是由周期内对应相位的时间特征进行线性变换得到,可以更好捕捉不同周期对应相位时间节点之间的联系.

(2)降低过拟合的可能性.这个优点源于小线性层的有效运用,极大地减少了线性头的参数量,使得模型更加轻量.以往的模型对于Transformer的输出并不在意其张量特性,只是将其铺平后用一个全连接层将其映射到需要的预测序列长度,这样使得线性头中的神经元个数最少也是六位数,大大增加了模型过拟合的可能.而小线性层在每个特征结点上分别使用多层感知机,对特征个数的维度进行线性变换,参数量大大减小.并且由于小线性层在一个周期长度的特征张量上使用相同的多层感知机,这使得线性头的参数量不会因为输出序列长度的增加而剧烈膨胀.对应实验结果见图9.

(3)从更长的回溯窗口学习的能力增强.简单地扩展输入序列长度 $L$ 是以更大的内存和计算使用为代价的.CycleLLH通过将复杂度与周期划分步长联系起来,便可以使得复杂度与 $L$ 几乎无关,这使得模型使用更长的回溯窗口进行学习时更具有潜力.通过实验证明CycleLLH在回溯窗口更大时,确实会得到更好的预测性能,对应详细实验结果见图10.

(4)降低了时间和空间复杂度.原始Transformer在时间和空间上都具有 $O(N^2)$ 复杂度,其中 $N$ 是输入令牌的数量.在没有预处理的情况下, $N$ 将具有与输入序列长度 $L$ 相同的值,这在实验和实践中成为计算时间和内存的主要瓶颈.通过应用周期整合,可以将 $N$ 减少为一个周期划分步长: $N=C$ ,随着输入序列长度的增加复杂度几乎不变,将二次复杂

度降低为一个恒定值.对应实验结果见图11.

(5)充分利用原始数据的特征.将经过周期整合得到的数据进行数据倍增后与编码器的输出进行拼接,然后一起经过线性层,这保证了原始数据对于预测结果有足够的影响.

### 3.2 模型架构

单变量网络流量预测考虑以下问题:给定长度为 $L$ 的单变量时间序列回溯窗口 $(x_1, x_2, \dots, x_L)$ ,其中每个时间结点 $t$ 的 $x_t$ 是维度为1的向量,希望预测未来的 $T$ 个值 $(x_{L+1}, x_{L+2}, \dots, x_{L+T})$ .

本文的CycleLLH架构如图3所示.模型使用普通的Transformer Encoder作为其核心架构,输入的单变量流量时间序列通过实例归一化进行传递,并按照周期化整合后进行数据倍增,最后将这些周期序列拼接作为编码器的输入令牌.对于单变量时间序列预测,输入令牌组成的张量具有序列长度和特征个数两个维度.

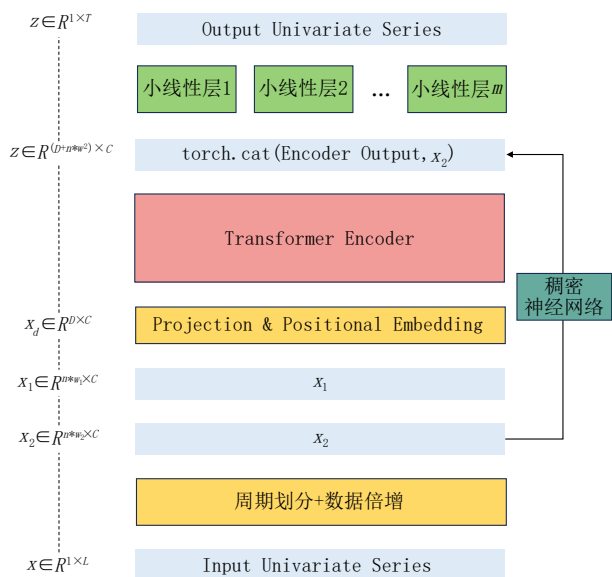


图3 CycleLLH模型结构

(1)周期整合.对于输入的单变量时间序列  $x \in R^{1 \times L}$  (若 $L$ 必须被周期划分步长 $C$ 整除),先根据其数据集周期划分为 $n$ 个长度为一个周期的时间序列.将其周期划分步长表示为 $C$ ,可以得到 $n$ 个  $x^{(i)} \in R^{1 \times C}$  的周期序列,  $n = \left\lceil \frac{L}{C} \right\rceil$ .然后将这 $n$ 个周期序列拼接得到长度为 $C$ 维度为 $n$ 的张量  $x \in R^{n \times C}$ .简易过程如图4所示.这意味着注意力矩阵的内存使用和计算复杂度只与当前数据集的周期划分步长有关,趋近于一个常数,而不会随着回溯窗口 $L$ 变化



而增大或减小. 因此降低受训练时间和GPU内存的限制.

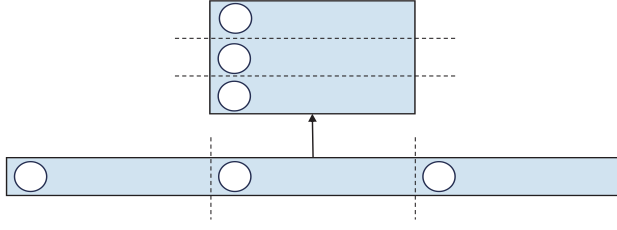


图4 输入序列具有3个周期进行周期整合的示意图

(2)输入数据倍增. 周期划分后得到的 $x \in R^{n \times C}$ , 并不直接作为编码器的输入, 而是将 $x$ 复制 $w_1$ 次, 在特征个数的维度上进行拼接得到 $x_1 \in R^{(n \times w_1) \times C}$ . 同样使用稠密神经网络结构, 将 $x$ 复制 $w_2$ 次, 在特征个数的维度上进行拼接得到 $x_2 \in R^{(n \times w_2) \times C}$ . 简易过程如图5所示. 这样做的目的: 少量增加后续线性层中的参数量, 可以学习到序列更多的特征; 充分利用原始特征序列中的信息, 增强原始数据对最终输出的影响.

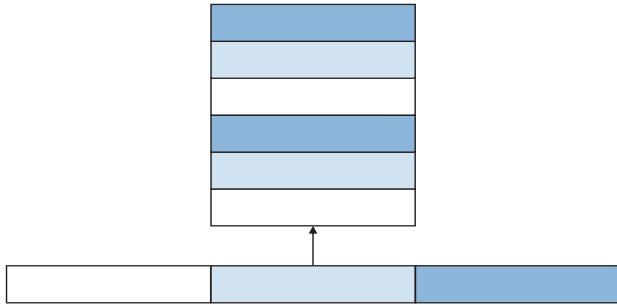


图5 复制一份拼接到特征个数维度的示意图

(3)Transformer Encoder. 本文使用一个普通的Transformer Encoder作为模型的主体, 其将观测到的信息映射到潜在的表示. 通过可训练的线性函数将周期整合和数据倍增后的输入序列 $x_1 \in R^{(n \times w_1) \times C}$ 映射到 $D$ 维 $W_c = R^{D \times (n \times w_1)}$ , 并应用可学习的位置编码 $W_{pos} \in R^{D \times C}$ 来检测输入序列的时间顺序,  $x_d = W_c x_c + W_{pos}$ .  $x_d \in R^{D \times C}$ 代表被输送到图3中Transformer Encoder的长度为 $C$ 的输入令牌. 多头注意力机制中的每一个头 $h=1, \dots, H$ 将会把输入分别转化为query矩阵 $Q_h = (x_d)^T W_h^Q$ , key矩阵 $K_h = (x_d)^T W_h^K$ , value矩阵 $V_h = (x_d)^T W_h^V$ , 其中 $W_h^Q, W_h^K \in R^{D \times d_k}$ ,  $W_h^V \in R^{D \times d_v}$ . 在这之后使用如下计算公式:

$$(O_h)^T = \text{Attention}(Q_h, K_h, V_h) =$$

$$\text{softmax}\left(\frac{Q_h K_h^T}{\sqrt{d_k}}\right) V_h$$

得到注意力矩阵 $O_h \in R^{D \times C}$ , 多头注意力块还包括BatchNorm层和带有残差结构的前馈神经网络. 最终其生成的表示为 $z \in R^{D \times C}$ .

(4)小线性层. 小线性层注重于周期内对应相位的一一对应映射. 将编码器的输出与数据倍增 $w_2$ 倍的输入序列 $x_2 \in R^{(n \times w_2) \times C}$ 进行拼接得到 $z \in R^{(D + n \times w_2) \times C}$ , 然后将其分别经过 $m$ 个多层感知机 $W_m = R^{1 \times (D + n \times w_2)}$ 得到 $m$ 个 $z^{(i)} \in R^{1 \times C}$ ,  $m$ 代表预测序列长度 $T$ 包含的周期数向上取整 $m = \left\lceil \frac{T}{C} \right\rceil$ . 将这 $m$ 个张量在序列长度的方向进行拼接, 然后截取所需要的预测长度, 得到最终的输出 $z = (\widehat{x_{L+1}}, \dots, \widehat{x_{L+T}}) \in R^{1 \times T}$ .

(5)损失函数. 本文主要选择用均方误差损失来衡量预测值和真实值之间的差异, 计算公式:

$$\mu = E_x [\widehat{x_{L+1:L+T}} - x_{L+1:L+T}]^2$$

(6)实例归一化. 这项技术是最近被提出用以缓解训练和测试数据之间的分布偏移效应<sup>[36]</sup>. 其简单地用零均值和单位标准差对输入时间序列实例 $x$ 归一化. 其实就是在周期整合前对输入时间序列 $x$ 进行归一化, 这个过程会得到输入序列的平均值和偏差值, 最后将这个平均值和偏差添加到小线性层的输出得到最终预测序列.

## 4 实验结果与分析

### 4.1 网络流量序列预测

本文主要在六个单变量数据集上评估了CycleLLH的性能, 包括: (1)NetTh和NetTm记录了移动运营商从2016. 1. 1到2017. 5. 1收集的流量数据, 数据来源为<https://github.com/JinScientist/traffic-data-5min>; (2) MilanM记录了米兰市从2013. 10. 31至2014. 1. 1的网络流量活动, 数据来源为<https://github.com/arunasubbiah/milan-telecom-data-modeling>; (3) CityAh、CityBh和CityCh记录了A、B、C三个不同地区从2017. 1. 至2018. 11. 15的每小时粒度网络流量数据, 数据来源为<https://github.com/Philxxw/DMNN>. 这些数据在过去几年已经被用于流量预测相关方面的研究<sup>[37-38]</sup>.

表1总结了这些数据集的统计数据,特征数是数据集每一个时间节点的维数;时间步长是每个数据集总的时间节点数;节点间隔是数据集每两个时间节点的采集时间间隔;每个数据集的周期划分步长 $C$ 是通过分析流量序列的自然周期并将 $C$ 设置为不同数值进行实验得到的,具体分析以及实验结果请见附录A,从表8中可以清楚地看到,对于NetTh数据集 $C=168$ 周期时间为一周效果最好,NetTm数据集 $C=48$ 周期时间为半天效果最好,MilanM数据集 $C=96$ 周期时间为一天效果最好,CityAh数据集 $C=24$ 周期时间为一天效果最好.

表1 用于实验的网络流量数据集的统计信息

数据集	特征数	时间步长	节点间隔	周期划分步长
NetTh	1	11 688	1 h	168
NetTm	1	46 752	15 min	48
MilanM	1	5952	15 min	96
CityAh	1	16 416	1 h	24
CityBh	1	16 416	1 h	24

本文最终选择了iTTransformer-2024<sup>[39]</sup>、PatchTST-2023<sup>[28]</sup>、RLinear-2023<sup>[40]</sup>、DLinear-2023<sup>[41]</sup>、TimesNet-2023<sup>[29]</sup>、Crossformer-2023<sup>[42]</sup>和Autoformer-2021<sup>[17]</sup>作为基准模型.下文若提到“基准模型”即为这七个模型.未使用网络流量预测的模型的原因有:未找到合适的公开代码,选用的时序预测领域已通过实验验证在能源、交通、经济、天气和电力等多个领域都有着十分优秀的预测性能,特别iTTransformer在网络流量数据集(比如支付宝在线交易分钟采样服务器负载)上也验证了其预测

性能.

由于本文提出的CycleLLH需要输入序列长度是当前周期划分步长的倍数,NetTh、CityAh、CityBh和CityCh的回溯窗口 $L=336$ ,NetTm和MilanM的回溯窗口 $L=384$ .但为了避免低估基准模型的预测表现,本文还针对四个不同的回溯窗口 $L\in\{96, 336, 384\}$ 运行iTTransformer、PatchTST、RLinear、DLinear、TimesNet、Crossformer、Autoformer,并选择最佳结果作为基准模型的实验数据.所有基准模型在NetTh、MilanM、CityAh、CityBh和CityCh中的实验设置都与基准模型对应论文中ETTh1实验设置相同,在NetTm中的实验设置都与基准模型对应论文中ETTm1实验设置相同.有关基准模型的更多实验数据,请参阅附录B.(ETTh1和ETTm1是时间序列预测模型中常用的数据集,包含2016年7月至2018年7月的七种石油和电力变压器的负载特征,PatchTST、Autoformer、Informer、Transformer均在评估了其模型性能,数据可以从<https://github.com/zhouhaoyi/ETDataset>获取.)

本文计算单变量流量序列预测的MSE(Mean Square Error,均方误差)和MAE(Mean Absolute Error,平均绝对误差)作为度量.表2展示所有模型在6个预测长度下MSE和MAE的均值,其中NetTh、NetTm、CityAh、CityBh和CityCh的预测长度 $T\in\{96, 168, 192, 240, 336, 720\}$ ,MilanM的预测长度 $T\in\{48, 96, 168, 192, 240, 384\}$ ,下文提到“6个预测长度”均和本次实验相同.最佳成绩以粗体显示,次佳成绩以下划线显示,模型在每个预测长度的具体数据参考附录C.

表2 CycleLLH和基准模型的预测结果均值

数据集	模型名称															
	CycleLLH		iTTransformer		PatchTST		RLinear		DLinear		TimesNet		Crossformer		Autoformer	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
NetTh	<b>0.292</b>	<b>0.330</b>	0.425	0.458	0.353	0.397	<u>0.333</u>	<u>0.376</u>	0.378	0.400	0.349	0.395	0.527	0.520	0.587	0.584
NetTm	<b>0.426</b>	<b>0.400</b>	0.690	0.610	0.491	0.462	0.613	0.525	0.630	0.530	<u>0.482</u>	<u>0.444</u>	0.509	0.516	0.740	0.628
MilanM	<b>0.038</b>	<b>0.130</b>	0.062	0.174	0.064	0.183	<u>0.058</u>	<u>0.165</u>	0.228	0.404	0.067	0.189	0.461	0.483	0.270	0.403
CityAh	<b>0.173</b>	<b>0.293</b>	0.195	0.330	0.232	0.361	<u>0.186</u>	<u>0.312</u>	0.218	0.349	0.221	0.344	0.499	0.509	0.534	0.594
CityBh	<b>0.303</b>	<b>0.354</b>	0.372	0.415	0.380	0.434	<u>0.329</u>	<u>0.386</u>	0.374	0.425	0.355	0.403	0.507	0.513	1.214	0.890
CityCh	<b>0.240</b>	<b>0.331</b>	0.286	0.381	0.295	0.387	<u>0.245</u>	<u>0.338</u>	0.303	0.395	0.302	0.386	0.504	0.511	1.302	0.923

CycleLLH中一共有三个超参数, $C$ 、 $w_1$ 、 $w_2$ . $C$ 为模型在当前数据集的周期划分步长. $w_1$ 为输入到编码器的网络流量序列数据倍增的倍数, $w_2$ 为附加到编码器输出的网络流量序列数据倍增的倍数.对

于每个数据集会进行多次实验,其中 $w_1\in[1,20]$ , $w_2\in[0,20]$ , $w_1$ 不能取0否则将没有数据输入到编码器, $w_2$ 取0则代表不使用稠密神经网络,取最优结果作为CycleLLH在当前数据集的表现.



所有实验都在PyTorch中实现,大部分实验都在单个NVIDIA 1060 6 GB上进行了,少数实验是在单个NVIDIA 1080Ti 11 GB上进行的,比如Autoformer在 $T=720$ 长度时. 对于CycleLLH我们使用ADAM作为优化器,L2Loss用于模型优化,初始学习率为 $10^{-4}$ ,最大迭代次数为60.

分析表2中的预测结果,本文的模型CycleLLH在所有模型中取得了最佳的预测性能,在六个数据集上都获得了远超第二名的最优MSE和MAE均值. 根据附录C中表11 CycleLLH和基准模型的完整预测结果,CycleLLH与基准模型所能提供的最

佳结果相比,在MSE和MAE标准下:

NetTh数据集上实现了23.3%、12.2%的总体降低;NetTm数据集上实现了7.6%、9.2%的总体降低;MilanM数据集上实现了37.4%、22.4%的总体降低;CityAh数据集上实现了5.5%、6.1%的总体降低;CityBh数据集上实现了8.1%、8.4%的总体降低;CityCh数据集上实现了1.8%、2.1%的总体降低.

同时本文将基准模型中表现较好的iTransformer、PatchTST和RLinear模型与CycleLLH的预测图像进一步分析比较. 图6展示四个模型在NetTh和CityAh的预测图像,图7展示了四个模型在NetTm

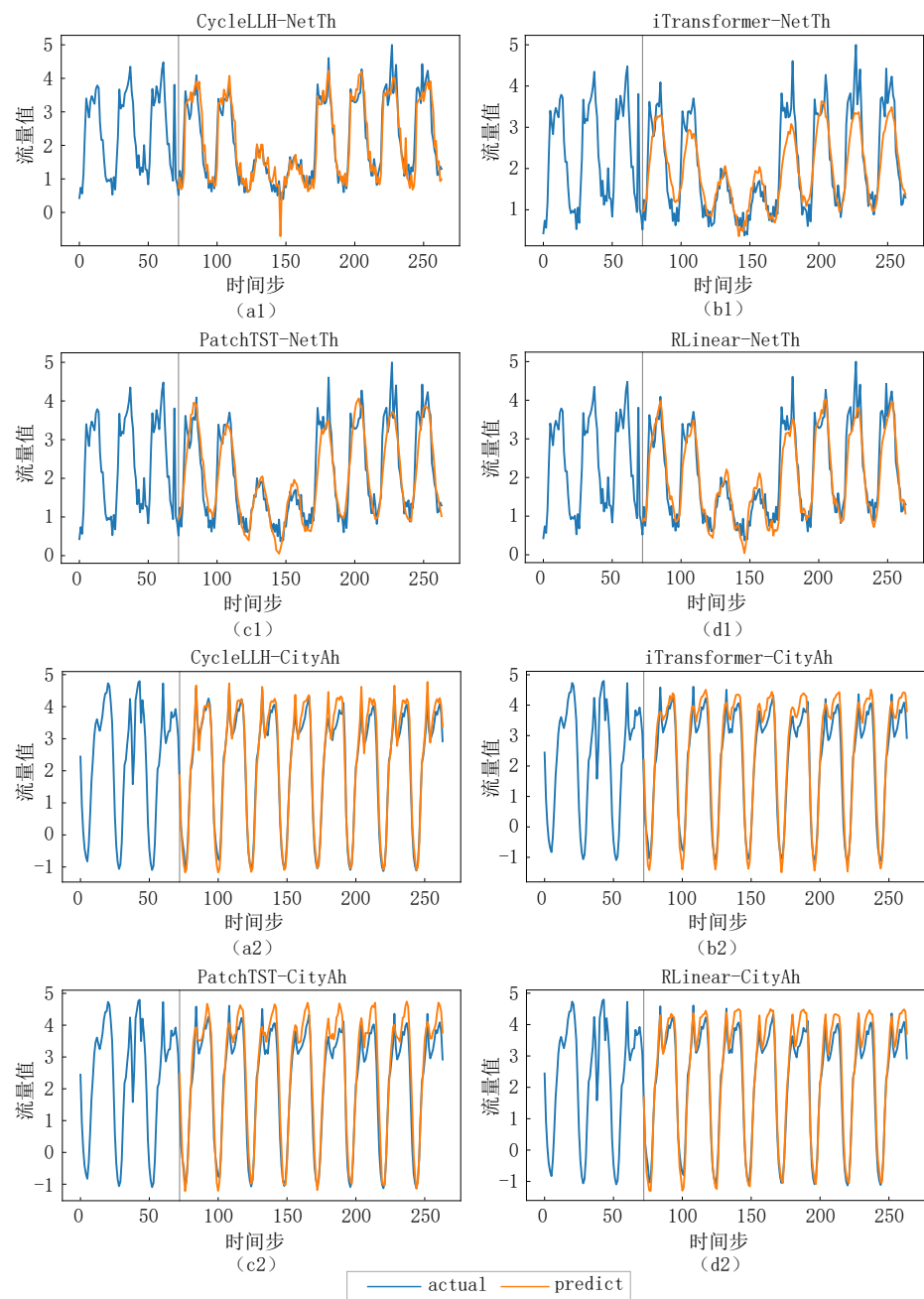


图6 NetTh和CityAh数据集上部分预测对比图像

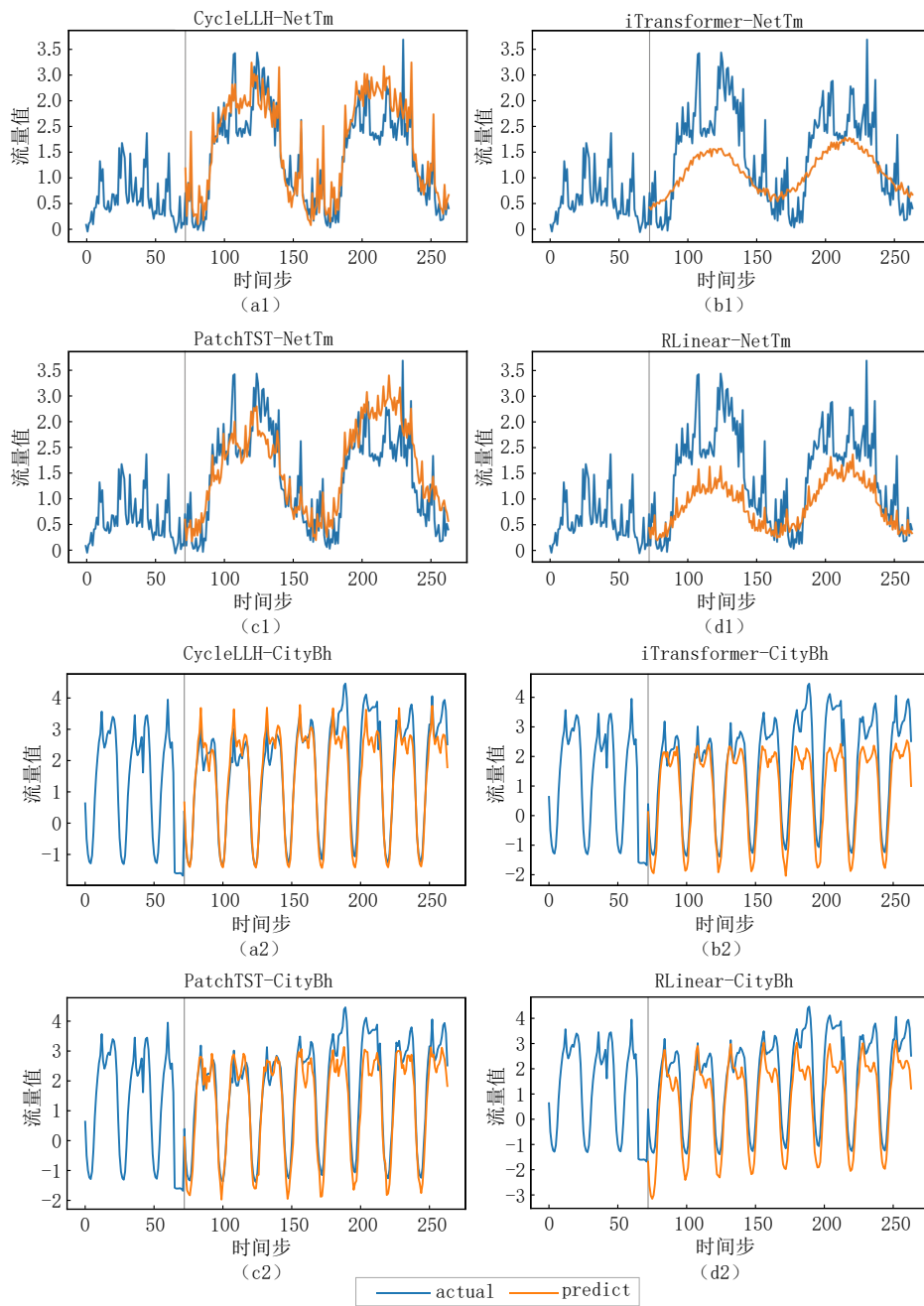


图7 NetTm和CityBh数据集上部分预测对比图像

和CityBh的预测图像,图8展示了四个模型在MilanM和CityCh的预测图像. 图的标题表示该图对应的模型与数据集,每个子图中的(a1)、(a2)代表CycleLLH模型,(b1)、(b2)代表iTransformer模型,(c1)、(c2)代表PatchTST模型,(d1)、(d2)代表RLinear模型. 图的横坐标是时间步长,图的纵坐标是在该时间对应的流量值,其中橙色是预测序列,蓝色是真实序列.

根据观察,无论是短时间还是长时间预测,这四个模型都能预测不同网络流量数据的基本走势. 然

而,CycleLLH明显在细节拟合方面表现得更为出色. 对于NetTh数据集,CycleLLH能够精准捕捉其低峰呈突刺状的特点;对于CityAh、CityBh、CityCh和MilanM数据集中极具规律的部分,CycleLLH甚至能做到完全拟合;而对于变化更错综复杂的NetTm数据集,CycleLLH在趋势把握上更精准、对部分流量突变预测也更准确.

## 4.2 消融研究

### 4.2.1 周期整合的作用

为了探究CycleLLH中的周期整合是否能提高

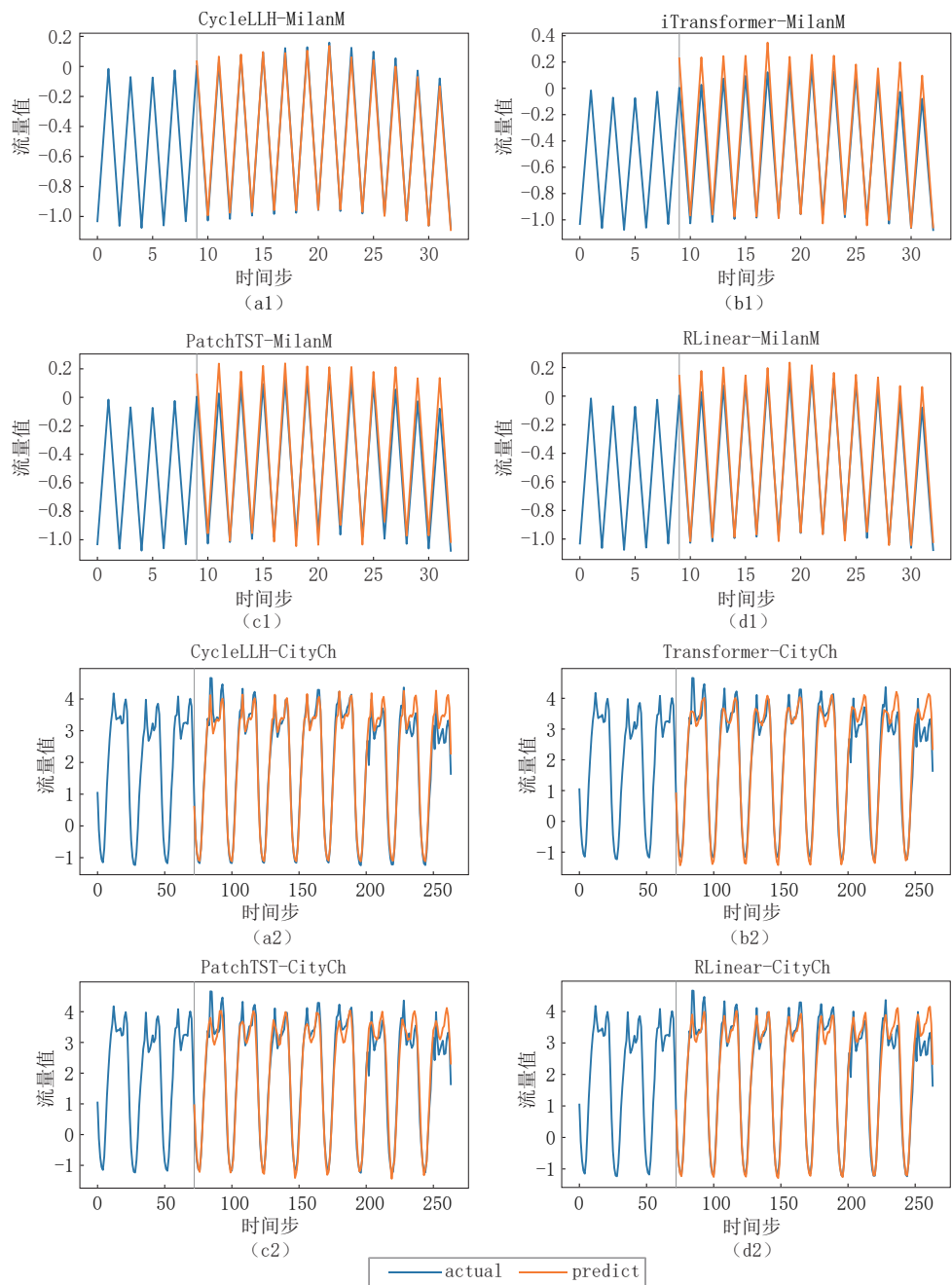


图8 MilanM和CityCh数据集上部分预测对比图像

模型在网络流量数据预测上的表现,我们将对“Patching+全连接层”(PatchTST)和“周期整合+全连接层”两种不同组合进行以下对比实验.

PatchTST的主要技术为Patching,其核心是将序列划分为一系列相同长度的段,有两个超参数 $seq\_len$ (每一个分段的长度)和 $stride$ (相邻两个段间隔的步长).为了探究Patching技术对于网络流量数据是否有效,实验一在不同分段长度以及段间隔下得到PatchTST在NetTh数据集上的表现.本文进行了210次回溯窗口大小 $L=336$ ,预测序列长度

$T=720$ 的实验,其中 $seq\_len \in [1, 20]$ ,对于每一个 $seq\_len, stride \in [1, seq\_len]$ .本实验即为表3中的第一行“Patching+全连接层”的情况,这210次实验在MSE标准下的最小损失值为0.2844,最大损失值为0.3908,平均损失值为0.3365,表4展示了 $seq\_len \in [1, 5]$ 的部分实验结果.

实验二为探究将Patching技术替换为本文的周期整合和数据倍增技术后在NetTh数据集上的整体表现.本文进行了420次回溯窗口大小 $L=336$ ,预测序列长度为 $T=720$ 的实验,其中 $w_i \in [1, 20]$ ,



$w_2 \in [0, 20]$ . 本实验即为表 3 中第二行“周期整合+全连接层”的情况, 这 420 次实验在 MSE 标准下的最小损失值为 0. 2848, 最大损失值为 0. 3239, 平均损失值为 0. 2921.

通过对比实验结果, 应用周期整合后, 平均损失值降低了 13. 2%, 表明模型的整体预测性能得到了显著提升. 此外通过表 4 中 PatchTST 在

NetTh 数据集上的部分实验结果, 还观察到随着  $seq\_len$  和  $stride$  越小, 模型的损失表现更好, 甚至当其都等于 1 时得到最优值. 然而, 此时 Patching 技术的分段核心思想并未得到充分体现, 而且在相同的回溯窗口下, 该模型的复杂度明显高于使用周期整合的情况. 这一现象间接表明, 在网络流量数据中应用 Patching 技术的适配度并不理想.

表 3 使用不同组件在 NetTh 数据集上的统计数据

模型组件	在 MSE 标准下的损失值		
	MIN	MAX	AVG
Patching+全连接层(PatchTST)	0. 2844	0. 3908	0. 3365
周期整合+全连接层	0. 2848	0. 3239	0. 2921
周期整合+小线性层(CycleLLH)	0. 2644	0. 3369	0. 2733

表 4 PatchTST 在 NetTh 数据集上的部分实验数据

$stride$ 的值	$seq\_len$ 的值				
	1	2	3	4	5
1	0. 291	0. 291	0. 284	0. 300	0. 287
2		0. 300	0. 299	0. 290	0. 295
3			0. 301	0. 302	0. 309
4				0. 316	0. 307
5					0. 324

4. 2. 2 关于线性头的讨论

经过周期整合后, 编码器将输出长度为一个周期划分步长的特征序列. 基于这一特点, 本文提出了小线性层的概念, 即通过在每个相位的时间特征单独使用多层感知机, 实现了编码器输出特征序列与最终预测序列之间的一一对应映射. 为了深入探究小线性层与传统的全连接层之间的差异, 我们将对“周期整合+小线性层”(CycleLLH)和“周期整合+全连接层”两种不同组合进行以下对比实验. 不使用小线性层结构进行消融实验的原因是, 小线性层并不是一种通用架构, 需要和前面的嵌入模块等相互配合才有意义, 无法简单套用在如 PatchTST 等其他模型上.

实验一为探究“周期整合+小线性层”在 NetTh 数据集上的整体表现, 本文进行了 420 次回溯窗口大小  $L=336$ , 预测序列长度  $T=720$  的实验, 其中  $w_1 \in [1, 20]$ ,  $w_2 \in [0, 20]$ . 本实验即为表 3 中的第三行“周期整合+小线性层”的情况, 这 420 次实验在 MSE 标准下的最小损失值为 0. 2644, 最大损失值为 0. 3369, 平均损失值为 0. 2733.

实验二为探究“周期整合+全连接层”在 NetTh 数据集上的整体表现, 即将 CycleLLH 中的小线性层替换为一个全连接层, 本文进行了 420 次回溯窗

口大小  $L=336$ , 预测序列长度  $T=720$  的实验, 其中  $w_1 \in [1, 20]$ ,  $w_2 \in [0, 20]$ . 本实验即为表 3 中的第二行“周期整合+全连接层”的情况, 这 420 次实验在 MSE 标准下的最小损失值为 0. 2848, 最大损失值为 0. 3239, 平均损失值为 0. 2921.

通过对比实验结果, 使用小线性层后, 最小损失值降低了 7. 2%, 这表明该方法在提高网络流量预测精度方面取得了显著突破. 同时, 平均损失值也降低了 6. 4%, 这一结果甚至优于使用一个全连接层时的最小损失值, 进一步证明了模型具有一定的鲁棒性.

图 9 展示了在保持其他配置恒定、仅改变预测长度时, 两种不同模型架构(即“周期整合+小线性层”与“周期整合+全连接层”)在模型大小上的变化趋势. 研究表明, 在所有预测的序列长度下, 采用小线性层的模型在大小上均显著小于全连接层模型. 并且随着预测长度的递增, 小线性层模型的尺寸增长速率远低于全连接层模型. 这一发现不仅验证了小线性层在减少模型参数数量方面的优势, 这有助于降低模型过拟合的风险, 并赋予了模型预测更长未来序列的潜力.

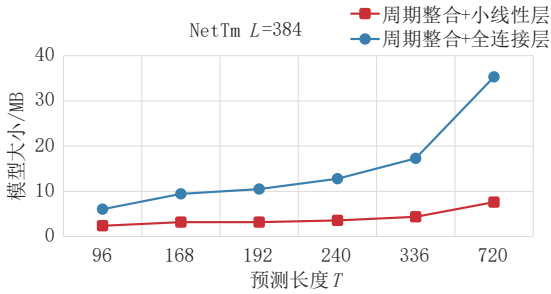


图9 使用小线性层/全连接层对模型参数数量的影响

#### 4.2.3 从更长回溯窗口学习的能力

从理论层面分析,如果模型能够利用更长的历史信息应能取得更佳的预测性能.然而,以往的研究指出,基于Transformer的模型的预测性能并不总是随着回溯长度的增加而提升<sup>[28,41]</sup>.这一现象可以归因于随着回溯长度的增长,输入到Transformer的令牌(tokens)数量也随之增加,从而可能导致模型在分配注意力时出现分散.然而,CycleLLH模型通过引入周期整合的机制,在确定了周期划分步长 $C$ 之后,无论回溯长度如何变化,输入到Transformer中的令牌数量都保持不变,这一设计巧妙地解决了注意力分散的问题.本实验旨在探索CycleLLH在网络流量数据集上是否具有从更长的回溯窗口学习的能力.

我们在NetTh、NetTm和CityAh预测序列长度 $T=96$ ,分别进行5次不同回溯窗口大小的实验,回溯窗口 $L \in \{48, 96, 192, 336, 672\}$ .同时使用PatchTST和iTransformer两个模型进行对比,以往研究已经通过实验证明其在电力和交通两个大型多变量数据集上具有从更长的回溯窗口中学习的能力.图10中的(a)、(b)和(c)分别展示三个模型在NetTh、NetTm和CityAh数据集上预测长度 $T=96$ 时,随着回溯窗口 $L$ 的增长MSE得分的变化图像.

分析图10可得,随着回溯窗口 $L$ 的逐步扩大,CycleLLH、iTransformer和PatchTST这三个模型在三个不同数据集上的MSE分数均呈现出稳步的下降趋势.这一趋势强有力地表明,这三个模型均能有效利用更长的回溯窗口所提供的信息,从而提升其预测性能.值得注意的是,在NetTm数据集的测试中相较于iTransformer和PatchTST模型,CycleLLH模型的MSE分数其下降趋势更为明显.这一发现进一步证实了CycleLLH模型在利用更长回溯窗口进行学习时,具备更强的适应性和学习能力.

图11展示了固定输出长度,改变回溯窗口大

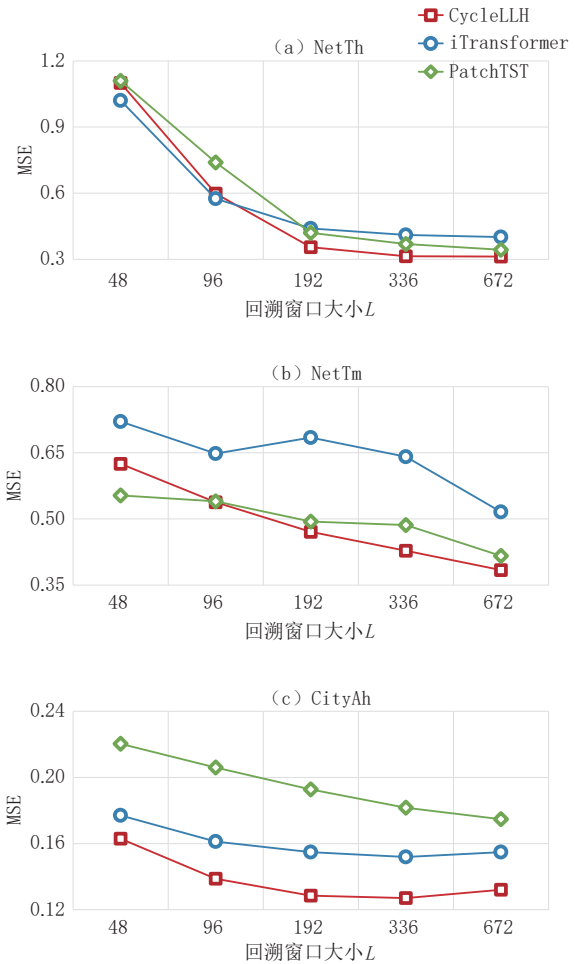
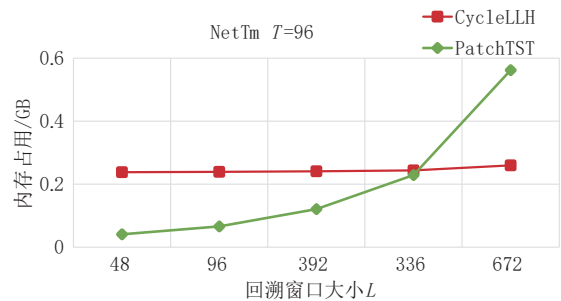
图10 回溯窗口 $L \in \{48, 96, 192, 336, 672\}$ ,固定预测长度 $T=96$ 的预测表现

图11 在不同回溯窗口下内存占用变化

小,CycleLLH和PatchTST模型在训练NetTm数据集时的内存占用变化趋势.可以观察到CycleLLH的内存占用几乎不变,而PatchTST的内存占用随着回溯窗口增大而增大.这说明了周期整合将输入令牌数量减少到与周期划分步长相同,有效降低了模型复杂度,将二次复杂度降低为一个恒定值.

#### 4.3 不同的周期划分对模型性能的影响

为了探究不同周期划分步长 $C$ 对模型性能的影响

响,我们在 NetTh 和 CityAh 数据集上,在回溯窗口  $L=336$ ,预测长度  $T=96$ ,针对  $C \in \{3, 4, 6, 7, 8, 12, 14, 16, 21, 24, 28, 42, 48, 56, 84, 112, 168\}$  进行了对比实验.

图 12 中的 (a) 和 (b) 分别展示 CycleLLH 在 NetTh 和 CityAh 数据集上,预测长度  $T=96$  时,随着周期划分步长  $C$  的变化 MSE 得分的变化图像. 其中虚线表示在当前数据集,相同的回溯窗口和预测序列长度设置下,表现最佳的基准模型 RLinear 的 MSE 值.

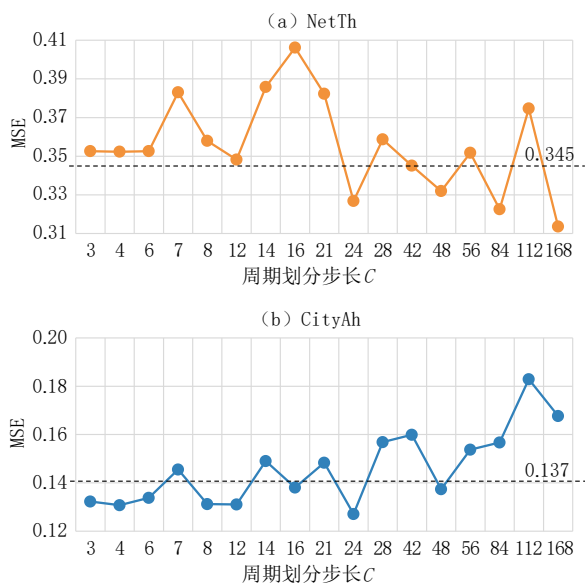


图 12 不同周期划分步长下模型的预测性能

分析图 12 可观察到,当  $C \in \{12, 24, 48, 84, 168\}$ ,即划分长度对应于半天、一天、两天等具有明确物理意义的时间周期时,模型展现出了更为优异的性能. 然而,令人惊喜的是,即使在  $C$  的取值不具备明确物理意义的情况下,CycleLLH 模型依然能够保持相对稳定的预测性能. 这一结果凸显了周期划分方法对于划分步长的高度容错性,即使在划分步长选择不够精确的情况下,也不会对模型的预测性能造成显著损害. 这一发现不仅为我们提供了在实际应用中灵活选择划分步长的可能性,也进一步证明了 CycleLLH 模型网络流量预测领域的强大鲁棒性和适应性.

根据上述实验结果,当数据集时间结点的具体含义不明确,无法通过常识或数学计算找到一个合适的周期划分步长  $C$  时,可以采用以下方法来寻找一个表现优秀的周期划分步长. 首先,通过观察序列图像,可以初步确定一个明显的周期,取一个明显

大于周期的划分步长  $C_1$ . 随后,通过逐步减少步长,可以在实验中找到一个表现最佳的划分步长  $C_2$ . 然而, $C_2$  可能并非最佳的划分步长,而是其倍数或约数. 因此,需要对  $C_2$  的倍数和约数进行多次实验验证. 最终,可以认为在这些实验中表现最佳的划分步长  $C$ .

#### 4.4 多层感知机不同层数对模型性能的影响

在初始阶段,针对小线性层中的多层感知机,本文仅采用了一层,将 Transformer 模块输出的特征序列中的特征个数维度直接映射到一维. 尽管这种方法在提升模型的预测性能上取得了一定的突破,具体表现为最小损失值相较于基准模型有所降低. 然而,当超参数  $w_1$  和  $w_2$  取不同值时,最大损失值和最小损失值之间出现了高达 32.8% 的差异,这表明模型的鲁棒性有待提升. 经过文献调研,了解到将高维数据一次性映射到低维,意味着模型必须在该层中丢弃大量信息,这很可能会对模型的性能产生负面影响. 因此,为了提高模型的鲁棒性和性能,本文有必要对多层感知机的层数进行更深入的优化和调整.

为了探究小线性层中多层感知机不同层数对模型性能的影响,本文在保持参数量大致相同的情况下,进行了大量的实验. 对于不同的层数,在 NetTh 数据集上进行了 420 次实验,回溯窗口大小  $L=336$ ,预测序列长度  $T=720$ ,其中  $w_1 \in [1, 20]$ ,  $w_2 \in [0, 20]$ . 模型在 MSE 标准下的最小损失值、最大损失值和平均损失值如表 5 所示.

分析表 5 可知,当层数较小时,随着层数的增加,最小、最大和平均损失值都逐渐减小,这表明模型的预测性能和鲁棒性得到了提升. 然而,当层数到达 6 以后,模型的性能提升并不明显. 综上所述,为了实现最佳的模型性能和鲁棒性,本文最终选择在小线性层中使用 6 层的多层感知机.

表 5 不同多层感知机层数在模型中的表现

层数	在 MSE 标准下损失值		
	MIN	MAX	AVG
1	0.2792	0.4156	0.3110
2	0.2674	0.3824	0.2902
3	0.2657	0.3387	0.2846
4	0.2653	0.3474	0.2812
5	0.2653	0.3446	0.2759
6	<b>0.2644</b>	0.3369	<b>0.2733</b>
7	0.2645	0.3517	0.2742
8	0.2653	0.3358	0.2737



4.5 鲁棒性分析

为了验证 CycleLLH 的鲁棒性,我们引入了一种简洁的白噪声注入方法. 具体而言,我们从原始的输入序列中挑选出一定比例的数据,比例为 $\epsilon$ ,随后在 $[-0.2X_i, 0.2X_i]$ 的范围内,对这些选中的数据施加扰动,其中 $X_i$ 表示原始数据. 接下来,我们将这些注入了噪声的数据用于模型的训练过程.

表6展示 CycleLLH 在不同扰动比例下,在各个数据集上6个预测长度的MSE和MAE得分均值,最后一列给出了相比于未注入噪声当前噪声比下MSE和MAE增长的百分比,增长越多说明模型的预测性能受损越严重.

如表6所示,随着扰动比例的增加,当噪声比例分别为1%、5%、10%时,MSE和MAE得分仅呈现为细微的上升趋势,增长的均值分别为1.1%、4.1%、9.5%. 这一结果表明,噪声的注入并未对模型的预测性能产生显著的负面影响,相反,它在一定程度上验证了 CycleLLH 在处理带有一定噪声数据时的鲁棒性. 我们的实验结果充分说明, CycleLLH 在面对较小噪声干扰的数据时,能够展现出卓越的稳定性,并且在有效应对各种异常数据波动方面,该模型展现出了显著的优势. 这一发现不仅验证了 CycleLLH 的健壮性,也为其在复杂、多变的数据环境中的应用提供了有力的支撑.

表6 CycleLLH 的稳健性实验(不同的  $\epsilon$  表示不同的噪声注入比例)

噪声比例	数据集												平均上升比例
	NetTh		NetTm		MilanM		CityAh		CityBh		CityCh		
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
$\epsilon=0\%$	0.292	0.330	0.426	0.400	0.038	0.130	0.173	0.293	0.303	0.354	0.240	0.331	
$\epsilon=1\%$	0.295	0.333	0.435	0.405	0.039	0.130	0.174	0.294	0.309	0.358	0.243	0.333	1.1%
$\epsilon=5\%$	0.302	0.343	0.440	0.411	0.040	0.132	0.187	0.305	0.322	0.368	0.251	0.339	4.1%
$\epsilon=10\%$	0.324	0.356	0.447	0.415	0.044	0.139	0.201	0.315	0.341	0.381	0.273	0.353	9.5%

4.6 归一化方法的合理性

为了严谨地验证归一化方法在单变量网络流量预测任务中的适用性和效果,我们比较了 CycleLLH、iTransformer 和 PatchTST 三种模型在

六个数据集上,使用归一化方法和未使用归一化方法时的MSE和MAE指标. 表7展示了模型在使用或未使用归一化方法时,在6个预测长度的得分均值,每个模型与自身比较,较好成绩以粗体显示.

表7 是否使用归一化实验结果对比,(+in)表示使用归一化,(-in)表示不使用归一化

数据集	模型名称											
	CycleLLH (+in)		CycleLLH (-in)		iTransformer (+in)		iTransformer (-in)		PatchTST (+in)		PatchTST (-in)	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
NetTh	<b>0.292</b>	<b>0.330</b>	0.406	0.403	<b>0.425</b>	<b>0.458</b>	1.488	0.923	<b>0.353</b>	<b>0.397</b>	0.553	0.522
NetTm	0.426	0.400	<b>0.422</b>	<b>0.370</b>	<b>0.690</b>	<b>0.610</b>	1.389	0.848	<b>0.527</b>	<b>0.493</b>	0.962	0.671
MilanM	<b>0.038</b>	<b>0.130</b>	0.135	0.320	<b>0.062</b>	<b>0.174</b>	0.269	0.401	<b>0.064</b>	<b>0.183</b>	0.386	0.571
CityAh	<b>0.173</b>	<b>0.293</b>	0.178	0.305	<b>0.195</b>	<b>0.330</b>	2.699	1.414	<b>0.232</b>	<b>0.361</b>	0.307	0.427
CityBh	<b>0.303</b>	<b>0.354</b>	0.357	0.411	<b>0.372</b>	<b>0.415</b>	4.090	1.630	<b>0.380</b>	<b>0.434</b>	0.486	0.511
CityCh	<b>0.240</b>	<b>0.331</b>	0.310	0.388	<b>0.286</b>	<b>0.381</b>	3.189	1.523	<b>0.295</b>	<b>0.387</b>	0.381	0.454

如表7所示,在采用归一化方法后,CycleLLH、iTransformer 和 PatchTST 的MSE和MAE指标均呈现出明显的下降趋势. 这一实验结果充分表明,归一化方法对于提升模型的预测性能具有积极作用,其能够有效地提高模型在单变量网络流量预测任务中的准确性. 因此,我们得出结论,归一化方法是单变量网络流量预测任务中一种合理且有效的数据处理手段.

并且我们观察到 CycleLLH 模型在未采用归一化预处理的情况下,其预测性能仅呈现微弱的下降

趋势. 尤为值得注意的是,在 NetTm 数据集上, CycleLLH 的预测能力甚至有所提升,这一发现突显了 CycleLLH 模型在应对不同数据集时的适应性与鲁棒性. 相比之下, iTransformer 和 PatchTST 模型在缺乏归一化处理的条件下,其预测性能出现了显著的下滑,这一对比进一步验证了 CycleLLH 模型在网络流量预测任务中的稳定性和优越性. 因此,从模型稳定性和预测效果的角度来看, CycleLLH 模型展现出了对于网络流量预测的极高适配性和应用价值.

## 5 总结与展望

本文旨在通过引入周期整合与小线性层两大核心设计,构建一种基于Transformer架构的高效网络流量预测模型,即CycleLLH模型.此模型不仅充分利用了网络流量数据固有的周期特性,而且在实践中被证实,相较于其他时间序列预测模型,在网络流量预测方面展现出了更为卓越的性能.特别地,周期整合的引入显著降低了模型的复杂度,增强了模型在较长回溯窗口下的学习能力,使其能够更有效地捕捉和解析数据中的周期性模式.

本文的核心思想在于探索并挖掘特定数据集的独特性质,这在处理特定预测任务时显得尤为重要.通过这一思想,我们提出的周期整合策略为处理具有周期性特征的数据集提供了一种新的视角和方法,其潜在价值在于为相关领域的研究和实践提供了一种基础性的数据处理与模型优化方法.

在深入探讨本研究所提出的模型时,必须认识到该模型仍存在一定的局限性.其中一个显著的局限在于,模型的输入序列长度 $L$ 必须能被周期划分步长 $C$ 整除.当面对无法整除的情境时,我们面临两种应对策略.首先,最直接的方法是舍弃最后不完整的周期序列,然而,这种方法不可避免地导致了信息利用率的降低,从而可能削弱模型的预测准确性.其次,我们尝试了利用插补技术来补充最后不完整的周期序列,使之成为一个完整的周期.然而,通过一系列的实验验证,我们发现使用前面完整周期序列的均值或最近周期对应相位的值进行插补,会对模型的预测性能产生显著的负面影响,甚至其效果不及直接舍弃不完整的周期序列.另一个局限性在于 $w_1$ 和 $w_2$ 将会带来二维搜索空间,在训练一个新的数据集时这可能会带来一些时间消耗.

尽管存在上述局限性,本文所构建的模型依然展现出其在现实场景中应用的巨大潜力.以某城市的电信运营商为例,他们发现夜间某些地区的流量需求相较于白天显著减少.基于这一发现,他们考虑通过关闭部分基站来降低能耗.然而,目前这一决策过程主要依赖于人工判断.若能够引入本文所提出的模型,通过人工智能对各基站流量进行预测,并在夜间智能地判断是否关闭基站,将能够显著提高资源利用效率,从而节省大量能源.我们也在积极推进这一项目.这一应用不仅体现了本模型的实际价值,也预示了其在未来智慧城市建设中的广阔前景.

## 参 考 文 献

- [1] Baldoni R, Caruso M, Cerocchi A, et al. Correlating power consumption and network traffic for improving data centers resiliency.arXiv preprint arXiv, 2014, 1405(2992): 1-6
- [2] Popescu D A, Zilberman N, Moore A. Characterizing the impact of network latency on cloud-based applications' performance. Technical Report of CAM, 2017, 914: 1-20
- [3] Jiang W. Cellular traffic prediction with machine learning: A survey. Expert Systems with Applications, 2022, 201 (117163): 1-19
- [4] Wang W, Zhou C, He H, et al. Cellular traffic load prediction with LSTM and Gaussian process regression//Proceedings of the ICC 2020-2020 IEEE International Conference on Communications. Dublin, Ireland, 2020: 1-6
- [5] Perveen A, Abozariba R, Patwary M, et al. Dynamic traffic forecasting and fuzzy-based optimized admission control in federated 5G-open RAN networks. Neural Computing and Applications, 2023, 35(33): 23841-23859
- [6] Zhang Q, Mozaffari M, Saad W, et al. Machine learning for predictive on-demand deployment of UAVs for wireless communications//Proceedings of the 2018 IEEE Global Communications Conference. Abu Dhabi, United Arab Emirates, 2018: 1-6
- [7] Gijon C, Toril M, Luna-ramirez S, et al. Long-term data traffic forecasting for network dimensioning in LTE with short time series. Electronics, 2021, 10(10): 1-19
- [8] Cui Y, Huang X, Wu D, et al. Machine learning-based resource allocation strategy for network slicing in vehicular networks. Wireless Communications and Mobile Computing, 2020, 2020(1): 1-10
- [9] Huang C W, Chen P C. Joint demand forecasting and DQN-based control for energy-aware mobile traffic offloading. IEEE Access, 2020, 8: 66588-66597
- [10] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need//Proceedings of the Advances in Neural Information Processing Systems.Long Beach, USA, 2017:1-15
- [11] Chen Z, Jiaye E, Zhang X, et al. Multi-task time series forecasting with shared attention//Proceedings of the 2020 International Conference on Data Mining Workshops.Sorrento, Italy, 2020: 917-925
- [12] Lin CY, Su HT, Tung SL, et al. Multivariate and propagation graph attention network for spatial-temporal prediction with outdoor cellular traffic//Proceedings of the 30th ACM International Conference on Information & Knowledge Management. Queensland, Australia, 2021: 3248-3252
- [13] Yu L, Li M, Jin W, et al. STEP: A spatio-temporal fine-granular user traffic prediction system for cellular networks. IEEE Transactions on Mobile Computing, 2020, 20 (12) : 3453-3466
- [14] Nie X, Zhou X, Li Z, et al. LogTrans: Providing efficient local-global fusion with transformer and CNN parallel network for

- biomedical image segmentation//Proceedings of the 2022 IEEE 24th Int Conf on High Performance Computing. Chengdu, China, 2022; 769-776
- [15] Kitaev N, Kaiser L, Levskaya A. Reformer: The efficient transformer//Proceedings of the International Conference on Learning Representations. Addis Ababa, Ethiopia, 2020; 1-12
- [16] Zhou H, Zhang S, Peng J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting//Proceedings of the AAAI Conference on Artificial Intelligence. Vancouver, Canada, 2021; 11106-11115
- [17] Wu H, Xu J, WANG J, et al. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting//Proceedings of the Advances in Neural Information Processing Systems. Online, 2021; 22419-22430
- [18] Zhou T, Ma Z, Wen Q, et al. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting//Proceedings of the International Conference on Machine Learning. Baltimore, Maryland, 2022; 27268-27286
- [19] Liu S, Yu H, Liao C, et al. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting//Proceedings of the International Conference on Learning Representations. Online, 2021; 1-20
- [20] Alvizu R, Troia S, Maier G, et al. Matheuristic with machine-learning-based prediction for software-defined mobile metro-core networks. *Journal of Optical Communications and Networking*, 2017, 9(9): D19-D30
- [21] Kirmaz A, Michalopoulos D S, Balan I, et al. Mobile network traffic forecasting using artificial neural networks//Proceedings of the 2020 28th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems. Nice, France, 2020; 1-7
- [22] Zhan S, Yu L, Wang Z, et al. Cell traffic prediction based on convolutional neural network for software-defined ultra-dense visible light communication networks. *Security and Communication Networks*, 2021, 2021(1): 9223965
- [23] Gao Y, Zhang M, Chen J, et al. Accurate load prediction algorithms assisted with machine learning for network traffic//Proceedings of the 2021 International Wireless Communications and Mobile Computing. Harbin, China, 2021; 1683-1688
- [24] Zhu Y, Wang S. Joint traffic prediction and base station sleeping for energy saving in cellular networks//Proceedings of the ICC 2021-IEEE International Conference on Communications. Montreal, Canada, 2021; 1-6
- [25] Swedha S, Gopi E S. LSTM network for hotspot prediction in traffic density of cellular network//Proceedings of the Machine Learning, Deep Learning and Computational Intelligence for Wireless Communication. Singapore, 2021; 35-47
- [26] Zeng Q, Sun Q, Chen G, et al. Attention based multi-component spatiotemporal cross-domain neural network model for wireless cellular network traffic prediction. *EURASIP Journal on Advances in Signal Processing*, 2021, 2021: 1-25
- [27] Wu Q, Chen X, Zhou Z, et al. Deep reinforcement learning with spatio-temporal traffic forecasting for data-driven base station sleep control. *IEEE/ACM Transactions on Networking*, 2021, 29(2): 935-948
- [28] Nie Y, Nguyen N H, Sinthong P, et al. A time series is worth 64 words: Long-term forecasting with transformers//Proceedings of the 11th International Conference on Learning Representations. Kigali, Rwanda, 2023; 1-24
- [29] Wu H, Hu T, Liu Y, et al. Timesnet: Temporal 2D-variation modeling for general time series analysis//Proceedings of the 11th International Conference on Learning Representations. Kigali, Rwanda, 2023; 22-44
- [30] Jia Y, Lin Y, Hao X, et al. Witran: Water-wave information transmission and recurrent acceleration network for long-range time series forecasting//Proceedings of the Advances in Neural Information Processing Systems. New Orleans, USA, 2023; 1-68
- [31] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998, 86(11): 2278-2324
- [32] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks//Proceedings of the Advances in Neural Information Processing Systems. Stateline, Nevada, USA, 2012; 1-7
- [33] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv*, 2014, 1409.1556; 1-14
- [34] Lin M, Chen Q, Yan S. Network in network//Proceedings of the International Conference on Learning Representations. Belgrade, Serbia, 2014; 1-10
- [35] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Seattle, USA, 2016; 770-778.
- [36] Ulyanov D, Vedaldi A, Lempitsky V. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv*, 2016, 1607.08022; 1-6
- [37] Huang J, Xiao M. Mobile network traffic prediction based on seasonal adjacent windows sampling and conditional probability estimation. *IEEE Transactions on Big Data*, 2020, 8(5): 1155-1168
- [38] Guo D, Xia X, Zhu L, et al. Dynamic modification neural network model for short-term traffic prediction. *Procedia Computer Science*, 2021, 187: 134-139
- [39] Liu Y, Hu T, Zhang H, et al. iTransformer: Inverted transformers are effective for time series forecasting//Proceedings of the International Conference on Learning Representations. Vienna, Austria, 2024; 1-14
- [40] Li Z, Qi S, Li Y, et al. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv*, 2023, 2305.10721; 1-12
- [41] Zeng A, Chen M, Zhang L, et al. Are transformers effective for time series forecasting? //Proceedings of the AAAI Conference on Artificial Intelligence. Pittsburgh, USA, 2022; 11121-11128
- [42] Zhang Y, Yan J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting//Proceedings of the 11th International Conference on Learning Representations. Kigali, Rwanda, 2023; 51-72



**附录 A.**

为了得到数据集合适的周期划分步长  $C$ , 我们首先绘制了六个数据集的序列图像如图 13 所示, 横坐标是时间步, 纵坐标是时间点对应的流量值, 图上方标题为该图像对应的数据集. 结合数据集两个时间节点采集的时间间隔, 我们初步认为流量序列的自然周期是一天, 对于 NetTh、CityAh、CityBh 和 CityCh 一天的时间步长为 24, 对于 NetTm 和 MilanM 一天的时间步长是 96. 同时我们还注意到对于 NetTh 一周内不同天也存在一定差异, 将一周作为其周期可能会更好, 于是为了得到更加每个数据集更合适的周期划分步长, 对于 NetTh 我们将  $C$  设为 24、48、168、336, 对于 NetTm 和 MilanM 我们

将  $C$  设为 24、48、96、192, 对于 CityAh 我们将  $C$  设为 12、24、48、168.

表 8 总结了 CycleLLH 在不同  $C$  设置下, 不同预测长度 MSE 和 MAE 得分的均值, 其中 NetTh、NetTm、CityAh、CityBh 和 CityCh 的预测长度  $T \in \{96, 168, 192, 240, 336, 720\}$ , MilanM 的预测长度  $T \in \{48, 96, 168, 192, 240, 384\}$ .

从表 8 可以清楚地看到, 对于 NetTh 数据集  $C=168$  周期时间为一周效果最好, NetTm 数据集  $C=48$  周期时间为半天效果最好, MilanM 数据集  $C=96$  周期时间为一天效果最好, CityAh、CityBh 和 CityCh 数据集  $C=24$  周期时间为一天效果最好.

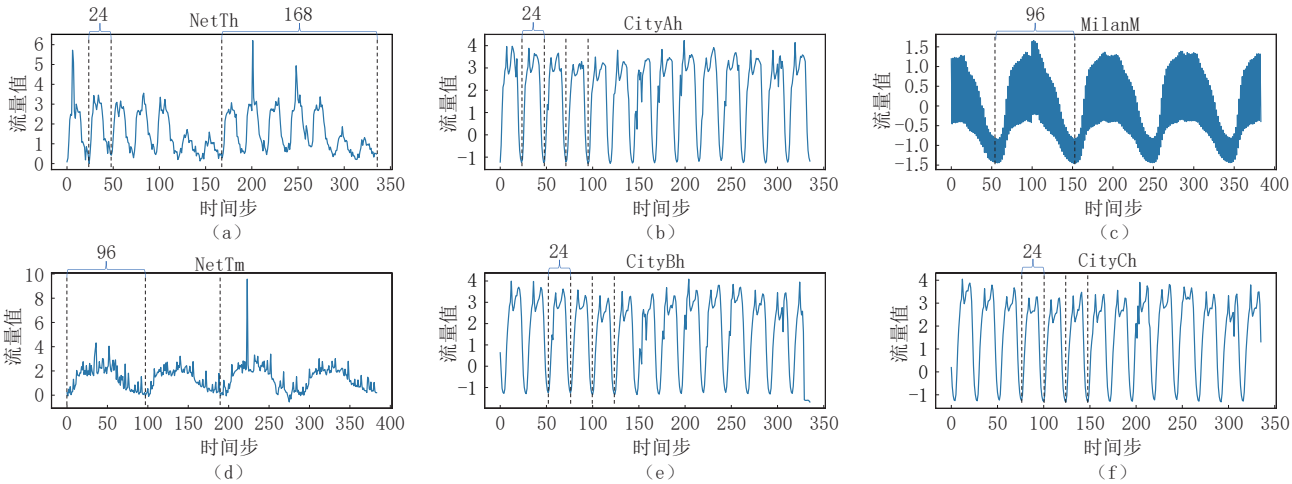


图 13 网络流量数据集的序列图像

表 8 六个数据集在不同周期划分步长  $C$  的实验结果

数据集	C 的设置							
	24		48		168		336	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
NetTh	0.3196	0.3664	0.3513	0.3907	<b>0.2924</b>	<b>0.3306</b>	0.4103	0.4196
	24		48		96		192	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
NetTm	0.4313	0.4042	<b>0.4265</b>	<b>0.4001</b>	0.4358	0.4060	0.5384	0.4609
MilanM	0.0528	0.1580	0.0526	0.1595	<b>0.0378</b>	<b>0.1267</b>	0.0443	0.1488
	24		48		96		192	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
CityAh	0.1750	0.2969	<b>0.1731</b>	<b>0.2928</b>	0.1814	0.3024	0.2072	0.3224
CityBh	0.3238	0.3685	<b>0.3031</b>	<b>0.3537</b>	0.3246	0.3712	0.3860	0.4007
CityCh	<b>0.2401</b>	<b>0.3304</b>	0.2405	0.3311	0.2567	0.3557	0.2983	0.3860

附录 B.

为了防止基线被低估并确保单变量原则,对于 iTransformer、PatchTST、RLinear、DLinear、TimesNet、Crossformer 和 Autoformer,在 NetTh、CityAh、CityBh 和 CityCh 运行了回溯窗口  $L \in \{96, 336\}$ ,在 NetTm 和 MilanM 运行了回溯窗口  $L \in \{96, 336, 384\}$ . 我们比较不同回溯窗口设置下的 MSE 和 MAE 分数,最终选择 MSE 均值最小的回溯窗口的实验结果作为基线.

表 9 给出了基准模型在各个数据集下不同回溯窗口设置,6 个预测长度的 MSE 和 MAE 均值,其中

NetTh、NetTm、CityAh、CityBh 和 CityCh 的预测长度  $T \in \{96, 168, 192, 240, 336, 720\}$ ,MilanM 的预测长度  $T \in \{48, 96, 168, 192, 240, 384\}$ ,最佳回溯窗口的实验结果用粗体显示. 表 10 总结了在 CycleLLH 和所有基准模型在各个数据集下使用的回溯窗口大小. 注意由于 MilanM 数据集的步长较短,当回溯窗口过大导致的训练集、验证集、测试集的数量减少会对实验结果产生很大影响,因此所有模型在 MilanM 数据集都选择  $L=384$  作为回溯窗口大小.

表 9 基准模型在不同回溯窗口的预测结果均值

数据集		模型名称													
		iTransformer		PatchTST		RLinear		DLinear		TimesNet		Crossformer		Autoformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
NetTh	96	0.524	0.519	0.636	0.574	0.577	0.533	0.627	0.560	0.486	0.494	<b>0.527</b>	<b>0.520</b>	0.677	0.611
	336	<b>0.425</b>	<b>0.458</b>	<b>0.353</b>	<b>0.397</b>	<b>0.333</b>	<b>0.376</b>	<b>0.378</b>	<b>0.400</b>	<b>0.349</b>	<b>0.395</b>	0.684	0.614	<b>0.587</b>	<b>0.584</b>
NetTm	96	0.883	0.680	0.581	0.502	0.792	0.610	0.850	0.614	<b>0.482</b>	<b>0.444</b>	0.522	0.520	0.747	0.611
	336	0.749	0.641	0.531	0.493	0.647	0.545	0.662	0.548	0.502	0.445	<b>0.509</b>	<b>0.516</b>	<b>0.740</b>	<b>0.628</b>
	384	<b>0.690</b>	<b>0.610</b>	<b>0.491</b>	<b>0.462</b>	<b>0.613</b>	<b>0.525</b>	<b>0.630</b>	<b>0.530</b>	0.491	0.450	0.516	0.520	0.786	0.655
MilanM	96	0.048	0.154	0.058	0.180	0.045	0.150	0.270	0.431	0.050	0.159	0.473	0.484	0.299	0.429
	336	0.053	0.160	0.057	0.173	0.050	0.154	0.249	0.423	0.062	0.181	0.452	0.477	0.229	0.388
	384	<b>0.062</b>	<b>0.174</b>	<b>0.064</b>	<b>0.183</b>	<b>0.058</b>	<b>0.165</b>	<b>0.234</b>	<b>0.413</b>	<b>0.067</b>	<b>0.189</b>	<b>0.461</b>	<b>0.483</b>	<b>0.270</b>	<b>0.403</b>
CityAh	96	0.202	0.332	0.249	0.373	0.197	0.320	<b>0.214</b>	<b>0.340</b>	0.229	0.348	0.523	0.518	0.721	0.677
	336	<b>0.195</b>	<b>0.330</b>	<b>0.232</b>	<b>0.361</b>	<b>0.186</b>	<b>0.312</b>	0.218	0.349	<b>0.221</b>	<b>0.344</b>	<b>0.499</b>	<b>0.509</b>	<b>0.534</b>	<b>0.594</b>
CityAh	96	<b>0.372</b>	<b>0.415</b>	0.444	0.461	0.380	0.407	0.487	0.499	0.379	0.414	0.524	0.518	1.458	0.975
	336	0.373	0.426	<b>0.380</b>	<b>0.434</b>	<b>0.329</b>	<b>0.386</b>	<b>0.374</b>	<b>0.425</b>	<b>0.355</b>	<b>0.403</b>	<b>0.507</b>	<b>0.513</b>	<b>1.214</b>	<b>0.890</b>
CityAh	96	0.298	0.388	0.341	0.418	0.283	0.368	0.531	0.588	0.327	0.405	0.529	0.521	<b>1.302</b>	<b>0.923</b>
	336	<b>0.286</b>	<b>0.381</b>	<b>0.295</b>	<b>0.387</b>	<b>0.245</b>	<b>0.338</b>	<b>0.303</b>	<b>0.395</b>	<b>0.302</b>	<b>0.386</b>	<b>0.504</b>	<b>0.511</b>	1.441	0.915

表 10 CycleLLH 和基准模型在各个数据集下使用的回溯窗口大小

数据集	模型名称							
	CycleLLH	iTransformer	PatchTST	RLinear	DLinear	TimesNet	Crossformer	Autoformer
NetTh	336	336	336	336	336	336	96	336
NetTm	384	384	384	384	384	96	336	336
MilanM	384	384	384	384	384	384	384	384
CityAh	336	336	336	336	96	336	336	336
CityBh	336	96	336	336	336	336	336	336
CityCh	336	336	336	336	336	336	336	96

附录 C.

表 11 CycleLLH 和基准模型的完整预测结果

数据集		模型名称															
		CycleLLH		iTransformer		PatchTST		RLinear		DLinear		TimesNet		Crossformer		Autoformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
NetTh	96	<b>0.314</b>	<b>0.337</b>	0.411	0.442	0.370	0.403	0.346	<u>0.376</u>	0.363	0.385	<u>0.345</u>	0.389	0.427	0.453	0.544	0.540
	168	<b>0.291</b>	<b>0.328</b>	0.436	0.464	0.350	0.394	<u>0.327</u>	<u>0.370</u>	0.368	0.390	0.352	0.402	0.482	0.489	0.582	0.578
	192	<b>0.296</b>	<b>0.328</b>	0.440	0.467	0.356	0.397	0.334	<u>0.374</u>	0.380	0.402	<u>0.333</u>	0.375	0.494	0.496	0.537	0.545
	240	<b>0.304</b>	<b>0.336</b>	0.441	0.470	0.362	0.401	<u>0.345</u>	<u>0.382</u>	0.373	0.396	0.351	0.400	0.497	0.507	0.639	0.636
	336	<b>0.285</b>	<b>0.327</b>	0.406	0.444	0.344	0.393	<u>0.329</u>	<u>0.377</u>	0.374	0.400	0.363	0.403	0.543	0.531	0.623	0.604
	720	<b>0.265</b>	<b>0.323</b>	0.416	0.458	0.333	0.395	<u>0.319</u>	<u>0.376</u>	0.408	0.429	0.348	0.399	0.717	0.642	0.600	0.604
NetTm	96	<u>0.434</u>	<b>0.415</b>	0.638	0.578	0.450	0.440	0.558	0.490	0.571	0.489	0.489	0.466	<b>0.402</b>	<u>0.447</u>	0.697	0.615
	168	<b>0.441</b>	<b>0.412</b>	0.706	0.624	0.507	0.481	0.651	0.552	0.669	0.557	0.458	<u>0.434</u>	<u>0.446</u>	0.476	0.713	0.618
	192	<b>0.433</b>	<b>0.401</b>	0.704	0.621	0.508	0.480	0.661	0.559	0.676	0.562	<u>0.469</u>	<u>0.437</u>	0.470	0.490	0.734	0.609
	240	<b>0.448</b>	<b>0.408</b>	0.710	0.622	0.515	0.475	0.660	0.554	0.684	0.565	0.501	<u>0.458</u>	<u>0.497</u>	0.507	0.706	0.600
	336	<b>0.403</b>	<b>0.384</b>	0.694	0.609	0.503	0.466	0.624	0.528	0.644	0.537	<u>0.491</u>	<u>0.456</u>	0.533	0.533	0.847	0.695
	720	<b>0.400</b>	<b>0.381</b>	0.687	0.604	0.466	0.427	0.527	0.465	0.537	0.467	<u>0.484</u>	<u>0.414</u>	0.705	0.642	0.746	0.630
MilanM	48	<b>0.017</b>	<b>0.086</b>	0.033	0.130	0.038	0.137	<u>0.027</u>	<u>0.114</u>	0.103	0.265	0.029	0.120	0.365	0.416	0.251	0.386
	96	<b>0.017</b>	<b>0.089</b>	0.038	0.141	0.043	0.148	<u>0.032</u>	<u>0.127</u>	0.136	0.319	0.045	0.151	0.406	0.444	0.241	0.381
	168	<b>0.029</b>	<b>0.118</b>	0.055	0.165	0.059	0.175	<u>0.049</u>	<u>0.153</u>	0.268	0.450	0.069	0.193	0.442	0.472	0.366	0.439
	192	<b>0.033</b>	<b>0.124</b>	0.060	0.173	0.061	0.179	<u>0.055</u>	<u>0.162</u>	0.260	0.443	0.073	0.201	0.488	0.500	0.242	0.399
	240	<b>0.042</b>	<b>0.142</b>	0.065	0.179	0.069	0.194	<u>0.066</u>	<u>0.181</u>	0.312	0.496	0.079	0.213	0.497	0.508	0.259	0.413
	384	<b>0.088</b>	<b>0.225</b>	0.119	0.255	0.115	0.262	<u>0.119</u>	<u>0.256</u>	0.324	0.504	0.109	0.253	0.565	0.557	0.259	0.402
CityAh	96	<b>0.127</b>	<b>0.247</b>	0.152	0.290	0.182	0.320	<u>0.137</u>	<u>0.267</u>	0.146	0.274	0.181	0.312	0.401	0.441	0.461	0.562
	168	<b>0.150</b>	<b>0.271</b>	0.182	0.321	0.202	0.338	<u>0.158</u>	<u>0.288</u>	0.173	0.301	0.204	0.332	0.462	0.483	0.553	0.616
	192	<b>0.155</b>	<b>0.275</b>	0.191	0.330	0.208	0.343	<u>0.164</u>	<u>0.294</u>	0.178	0.305	0.202	0.328	0.457	0.481	0.481	0.569
	240	<b>0.173</b>	<b>0.295</b>	0.186	0.321	0.225	0.359	<u>0.182</u>	<u>0.312</u>	0.203	0.331	0.218	0.343	0.480	0.499	0.440	0.545
	336	<b>0.198</b>	<b>0.321</b>	0.207	0.341	0.252	0.381	<u>0.207</u>	<u>0.336</u>	0.223	0.351	0.246	0.366	0.536	0.533	0.500	0.581
	720	<b>0.236</b>	<b>0.348</b>	<u>0.250</u>	<u>0.374</u>	0.320	0.422	0.271	0.375	0.363	0.481	0.279	0.384	0.659	0.619	0.766	0.690
CityBh	96	<b>0.250</b>	<b>0.308</b>	0.311	0.373	0.326	0.394	<u>0.280</u>	<u>0.340</u>	0.300	0.354	0.310	0.364	0.402	0.440	1.033	0.833
	168	<b>0.276</b>	<b>0.332</b>	0.347	0.398	0.354	0.415	<u>0.304</u>	<u>0.364</u>	0.343	0.397	0.325	0.380	0.445	0.473	0.995	0.820
	192	<b>0.284</b>	<b>0.339</b>	0.358	0.407	0.362	0.421	<u>0.311</u>	<u>0.371</u>	0.349	0.403	0.342	0.394	0.468	0.487	1.043	0.843
	240	<b>0.299</b>	<b>0.354</b>	0.363	0.405	0.381	0.435	<u>0.330</u>	<u>0.389</u>	0.397	0.449	0.349	0.401	0.474	0.494	1.000	0.823
	336	<b>0.326</b>	<b>0.376</b>	0.394	0.430	0.409	0.454	<u>0.354</u>	<u>0.409</u>	0.427	0.473	0.373	0.421	0.533	0.533	1.002	0.828
	720	<b>0.384</b>	<b>0.414</b>	0.459	0.478	0.450	0.483	<u>0.397</u>	<u>0.441</u>	0.431	0.474	0.429	0.460	0.719	0.649	2.211	1.192
CityCh	96	<u>0.220</u>	<b>0.312</b>	0.245	0.351	0.268	0.369	<b>0.219</b>	<u>0.319</u>	0.231	0.331	0.264	0.358	0.408	0.445	0.892	0.778
	168	<b>0.232</b>	<b>0.323</b>	0.273	0.374	0.288	0.382	<u>0.236</u>	<u>0.330</u>	0.262	0.359	0.289	0.377	0.447	0.475	1.169	0.881
	192	<b>0.240</b>	<b>0.330</b>	0.282	0.380	0.296	0.387	<u>0.243</u>	<u>0.335</u>	0.307	0.401	0.306	0.387	0.457	0.482	1.506	0.990
	240	<b>0.255</b>	<b>0.340</b>	0.286	0.380	0.311	0.397	<u>0.259</u>	<u>0.347</u>	0.351	0.434	0.301	0.388	0.474	0.493	1.230	0.897
	336	<b>0.259</b>	<b>0.345</b>	0.309	0.395	0.313	0.398	<u>0.265</u>	<u>0.351</u>	0.369	0.446	0.339	0.415	0.533	0.530	1.322	0.915
	720	<b>0.237</b>	<b>0.337</b>	0.323	0.406	0.293	0.390	<u>0.248</u>	<u>0.347</u>	0.298	0.401	0.312	0.393	0.706	0.642	1.692	1.076

附录 D.

为了探究 CycleLLH 在 NetTh 数据集上的整体表现,本文进行了 420 次回溯窗口大小  $L=336$ , 预测序列长度  $T=720$  的实验,其中  $w_1 \in [1, 20]$ ,  $w_2 \in [0, 20]$ . 本实验中的表 12 即为表 3 中的第三行“周期整合+小线性层”的完整数据. 这 420 次

实验在 MSE 标准下的最小损失值为 0.2644,最大损失值为 0.3369,平均损失值为 0.2733. 同时可以注意到较好的 MSE 成绩总出现在  $w_2 \neq 0$  时,这证明稠密神经网络将原始数据拼接 to 编码器输出是有效的.

表 12 CycleLLH 在 NetTh 数据集上的实验数据

$w_2$ 的值	$w_1$ 的值																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	289	315	270	274	275	284	276	289	270	271	281	271	269	271	273	287	274	269	266	269
1	307	281	266	270	285	269	267	267	272	266	269	266	270	281	270	275	267	266	271	270
2	273	272	268	269	269	304	269	292	270	266	267	267	269	268	266	268	280	300	317	306
3	276	270	268	271	267	265	268	274	268	266	266	333	267	300	270	267	265	269	264	302
4	271	271	267	268	273	266	268	277	267	268	266	268	267	269	326	268	267	266	266	266
5	276	268	268	267	273	267	269	269	267	267	269	268	268	277	266	266	267	314	312	268
6	310	305	267	267	269	267	270	267	266	270	268	270	270	270	270	267	270	266	271	267
7	274	272	269	268	268	269	268	268	268	269	268	267	269	272	269	268	267	267	268	268
8	277	273	268	269	269	268	269	268	268	267	269	269	271	280	267	268	268	267	265	329
9	331	277	272	270	268	269	268	287	268	269	269	267	270	333	268	268	269	269	268	269
10	281	275	273	270	268	290	270	272	270	271	270	270	269	270	269	272	270	267	270	267
11	278	274	268	272	268	268	270	270	271	270	271	270	271	273	272	272	272	270	269	270
12	282	276	274	270	270	270	269	273	271	268	310	271	271	273	272	269	267	267	267	267
13	280	276	272	272	271	271	269	288	268	272	292	270	271	337	268	269	271	271	271	270
14	282	275	276	271	267	285	273	272	269	271	273	274	274	274	272	271	270	270	270	270
15	278	276	272	270	272	268	271	273	269	271	272	273	271	271	269	271	269	272	270	270
16	286	278	272	270	269	270	270	274	270	270	273	270	273	272	273	270	269	273	271	271
17	282	276	272	271	268	280	272	274	270	270	271	273	272	287	273	273	270	271	273	293
18	282	279	271	271	267	275	271	275	270	273	272	273	269	275	271	271	272	270	270	273
19	282	274	273	269	269	271	274	275	271	271	301	274	272	274	269	274	269	271	268	273
20	287	276	274	274	272	271	273	274	270	271	272	274	275	274	274	275	272	272	269	272



TANG Wen-Jie, M. S. candidate.

His research interest is time series prediction.

KONG Xiang-Yu, Ph. D. candidate. His research interest

is network situational awareness.

QI Heng, Ph. D., professor. His research interests

include new generation of Internet, and data center networking.

LIU Xiu-Long, Ph. D., professor. His research interests

include IoT and blockchain.

LI Ke-Qiu, Ph. D., professor. His research interests

include wireless networks, cloud computing, and software-defined data center networks.

XIAO Yi-Lei, Ph. D. candidate. His research interest is time series prediction.

Background

The occurrence of downtime events in current cloud services and cloud systems is not uncommon. One of the key factors is that cloud service providers lack accurate estimation and prediction of various application service traffic, which makes it difficult for them to timely reserve or dynamically allocate resources effectively, resulting in service crashes. If network traffic changes could be predicted in advance and appropriate preventive measures taken, more robust services could be provided to users. In short, we need to build an efficient and reliable network traffic prediction model. Time

series prediction has always been a hot topic in the field of artificial intelligence, and various time series prediction models based on Transformer have emerged in recent years. It is possible to migrate these models directly to the network traffic prediction problem, but the general model does not take into account the specific characteristics of a particular dataset.

Therefore, based on the concept of exploring the unique characteristics of datasets, this paper proposes a CycleLLH model according to the periodic characteristics of network traffic data. CycleLLH has two key designs: Cycle Integration divides the input flow sequence according to cycle, and



integrates them in the dimension of input tokens one-to-one; Little Linear Layer connects the output of the encoder and the prediction sequence in a one-to-one correspondence within the period. The model has the advantages of making full use of the periodicity of data, reducing the complexity of time and space, reducing the possibility of overfitting and making full use of the features of the original data. The experiment has proved that the cycle integration makes the model have stronger learning ability in the longer look-back window. The model proposed in this paper outperforms other general time series prediction

models in network traffic data, with an improvement of 12.3%, 8.4%, 29.9%, 5.8%, 8.3% and 2.0% on the six network traffic data sets, respectively.

This work is supported by the National Natural Science of China under Grant 62072069. This project aims to make advances to AI-driven network situational awareness including network traffic classification and prediction. This paper summarizes the research progress of network traffic prediction, then designing a new AI model for traffic prediction, which is more suitable for network traffic.