Computer Science and Application 计算机科学与应用, 2026, 16(1), 182-197
Published Online January 2026 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2026.161015

Hans汉斯

关注周期－趋势－波动的云平台任务数量预测方法

姜昕怡[1,2]，曾国荪[1,2*]

[1]同济大学计算机科学与技术学院，上海
[2]国家高性能计算机工程技术中心同济分中心，上海

收稿日期：2025年12月7日；录用日期：2026年1月9日；发布日期：2026年1月20日

摘　要

在云计算数据中心，准确预测未来一段时间内的用户任务达到数量并及时调整资源分配，对提升资源利用率和降低能耗十分重要。为应对云用户任务提交模式的高度动态性，以及长期时间序列预测过程中易产生误差累积的问题，提出了一种关注周期、趋势、波动特征的云平台任务数量预测方法。该方法首先从原始时间序列中提取单个可循环的周期模式，以便后续显式地利用时间序列中的稳定周期模式；然后引入周期注意力机制，通过设定周期长度对子序列进行聚合以提升周期模式的表达能力，并结合门控机制动态调节周期模式对预测结果的影响。最后，利用Transformer对去除周期分量的趋势－波动分量进行建模，实现了云任务数量的预测。利用Alibaba-cluster-gpu-v2020数据集，开展实验验证，实验结果表明该方法在长期预测精度和稳定性方面优于对比的基线方法，具有较好的实际应用价值。

关键词

云计算平台，任务到达数量，周期模式，趋势和波动，预测方法

A Cloud Platform Tasks Number Prediction Method Focusing on Period-Trend-Fluctuation

Xinyi Jiang[1,2], Guosun Zeng[1,2*]

[1]Department of Computer Science and Technology, Tongji University, Shanghai
[2]Tongji Branch, National Engineering & Technology Center of High Performance Computer, Shanghai

Received: December 7, 2025; accepted: January 9, 2026; published: January 20, 2026

*通讯作者。

文章引用: 姜昕怡, 曾国荪. 关注周期-趋势-波动的云平台任务数量预测方法[J]. 计算机科学与应用, 2026, 16(1): 182-197.
DOI: 10.12677/csa.2026.161015

Abstract

In cloud data centers, accurately forecasting the number of arriving user tasks over a future period and timely adjusting resource allocation are essential for improving resource utilization and decreasing energy consumption. To address the high dynamism of cloud users' task submission patterns and the issue of error accumulation commonly encountered in long-term forecasting, this paper proposed a method for cloud task arrival prediction with Periodic, Trend, and Fluctuation features. The method first constructed a single recurrent periodic pattern from the original time series to explicitly capture stable periodic behaviors. Meanwhile, the paper proposed a Period-Attention mechanism that aggregates long-term subseries through the explicit period length, thereby enhancing the representation capability of periodic components. Additionally, a gating mechanism is incorporated to dynamically regulate the influence of periodic information on the prediction results. Finally, the Transformer architecture was utilized to method and forecasted the Trend-Fluctuation components after removing the periodic components, achieving the prediction of the number of cloud tasks. Experiments are conducted on the Alibaba Cluster Trace GPU v2020 dataset and the results demonstrate that the proposed method consistently surpasses several baseline models in both long-term forecasting accuracy and stability, showing strong potential for practical application in cloud task prediction scenarios.

Keywords

Cloud Computing Platform, The Number of Tasks Arriving, Periodic Pattern, Trend and Fluctuation, Prediction Method

1. 引言

随着云计算技术的发展，越来越多的企业和开发者选择使用云服务器来托管应用程序和服务。据云计算白皮书(2024 年)显示，我国大型企业上云率超过 80%，已从“资源上云”向“深度用云”阶段发展，各类核心业务系统已逐步迁移到云端[1]。亚马逊云、谷歌云、阿里云等云服务提供商建设和运营了大量数据中心，为云用户提供多样化的云计算服务，并设计了多种计费方式以适配不同用户的使用需求。例如，在面向短期需求、业务负载波动大或难以预估资源需求的应用场景中，用户可选择灵活性高的按量付费计费方式。特别地，对于有长期稳定需求或可预估资源使用周期的业务场景，用户可选择成本更低的预留实例计费方式。

然而，在实际使用过程中，云计算平台往往难以准确掌握云用户应用程序的资源使用特征。由于应用程序的资源需求随时间变化，其具有高度动态性和复杂性，云平台无法准确预测未来资源使用情况时，容易出现资源配置决策失误，这往往会导致资源过度配置或配置不足，进而造成云数据中心资源浪费、成本低效等问题，甚至云服务提供商可能会违反与云用户签订的服务级别协议 SLA。因此，对云数据中心而言，根据用户历史数据对未来较长一段时间内的云计算任务数量进行精准地预测十分必要。云计算任务数量是决定云平台资源需求和业务负载的重要指标[2]，其变化趋势将直接影响云资源分配和负载均衡策略的制定。通过对未来到达任务数量的准确预测，云服务提供商可提前数小时、数天甚至数周满足用户的资源需求，避免违反服务级别协议，并使用先进的资源管理技术来提升服务质量 QoS，同时提高

资源利用率，降低硬件设施的能耗和运营成本。

在云计算任务预测中，长期时间序列(多步)预测比短期时间序列(单步)预测更有效，因为长期时间序列预测能够为云服务提供商提供更全面的未来趋势信息，从而支持更复杂的任务调度和资源配置策略。然而，与短期时间序列预测相比，长期时间序列预测更具挑战性。随着预测时间跨度的增加，时间序列的相关性逐步减弱，模型的预测性能会逐步下降。另外，后期的预测值依赖于前期的预测值，预测误差在迭代过程中逐步累积，对长期预测精度产生显著的负面影响。与此同时，云计算工作负载的动态性和复杂性增加了负载预测的难度。用户行为与资源需求的突发性变化会引入明显的负载波动，使时间序列呈现出更强的非平稳性特性，进一步干扰模型对潜在模式的学习，导致预测精度下降。

为缓解长期时间序列预测中预测性能难以保持稳定的问题，本文提出了一种关注周期、趋势、波动特征的方法，用于预测云计算用户任务到达数量。其主要思想是通过分析云计算任务数量的时间序列，将其分解为周期分量、趋势－波动分量。对其固有的周期成分进行显式建模，在后续生成多步预测结果时可直接使用已建模的周期分量，模型无需重复学习其周期特征和预测周期分量，只需预测趋势－波动分量，使得长期预测更加稳定。此外，本文提出了一种周期注意力机制，通过明确的周期长度聚合子序列能够更精准地建模周期分量，并内置了门控机制用于调整注意力分数对其输出的影响。最后，采用Transformer [3]架构来预测趋势－波动部分，其中解码器采用生成式解码器一步生成目标长度的预测结果，进一步减少动态解码过程累计误差的扩散。

2. 相关工作

针对云数据中心面临的资源利用率低和能耗浪费等问题，已有大量文献研究云计算负载预测。云计算用户任务到达数量是决定云计算负载的重要组成部分，主要聚焦于单位时间内任务达到数量的准确预测。云计算任务数量序列可视为一个时间序列，即在均匀时间间隔内连续测量的数据点集合。目前云计算负载预测的研究工作可分为基于传统统计学模型、基于经典机器学习方法和基于深度学习模型等。

基于统计学模型的负载预测方法模型简单、计算便捷，易于实现。自回归 AR、移动平均 MA 和自回归差分移动平均模型 ARIMA 都属于传统统计学模型。Calheiros 等人[4]提出了一种基于 ARIMA 模型的云工作负载预测模型，通过对实际 Web 请求数据的建模预测未来资源需求，从而提前分配或释放资源，以在保证 QoS 的同时提升资源利用效率。Bi 等人[5]提出了一种结合小波分解和 ARIMA 模型的混合方法，用于预测云数据中心到达任务数量。该方法利用 Savitzky-Golay 滤波对任务时间序列进行平滑处理，并借助小波分解将其分解为多个组成部分，针对趋势和各组成部分的统计特性分别建立 ARIMA 模型，再借助小波重构将各部分的预测结果进行整合。然而，传统统计学模型要求数据必须具有平稳性，即序列的均值和方差不发生明显变化，显然，云工作负载的时间序列不符合其平稳性的要求，因此传统统计学模型难以有效地建模变量中的复杂非线性关系。

基于经典机器学习的负载预测方法训练参数较少，所花费的计算开销较少，在数据集较小时更具优势，包括线性回归 LR、支持向量机 SVM、决策树和极限学习机 ELM 等模型。Islam 等人[6]采用了误差修正神经网络和 LR 来预测未来资源需求的激增，以可控的方式处理突发的工作负载。Zhongda 等人[7]提出了一种基于小波变换和多模型融合的预测方法，对趋势成分使用最小二乘支持向量机，对随机成分使用自回归集成移动平均模型，并采用 Gauss-Markov 算法对多个预测模型预测值进行融合。S. Jeddi 等人[8]提出了一种混合小波时间序列分解器和 GMDH-ELM 集成方法，在不同的时间频率尺度上将工作负载序列分解为三层，并使用特定的非线性 GMDH 模型对每个层次进行预测，通过 ELM 将来自不同层次的时间序列预测值结合起来。Kumar 等人[9]提出了一种基于 ELM 的工作负载预测模型，用于预测由季节性加法分解方法获得的趋势成分、季节性成分和随机成分，减少其复杂的非线性行为，提高了预测精

度并大幅缩短训练时间。Feng 等人[10]提出了一个面向动态云工作负载预测的集成模型，由预测宏观趋势、季节性成分的 Macro-SARIMA 模型和捕捉微观变化的 Micro-ASW-GBDT 模型这两个局部预测器组成。其中，Micro-ASW-GBDT 引入了一种自适应滑动窗口算法，动态调整窗口大小以提升在线回归的精度，并提出了时间局部性集成策略，将局部预测器行为建模为多类回归问题。相较于传统统计学模型，经典机器学习模型虽然能处理非线性和高维数据，具有更强大的学习能力，但是存在对数据质量敏感、容易过拟合等问题，在处理大规模复杂数据时通常需要大量的人工特征工程；此外，在面对高度动态或非线性关系复杂的数据时，模型的泛化能力和预测精度可能受限，难以自动适应数据的变化。

基于深度学习的负载预测方法在处理复杂庞大的数据时表现出更强的建模能力，通过增加神经网络的层数和训练参数能够有效地提升对非线性关系的拟合能力，从而提高预测精度，包括循环神经网络 RNN、长短期记忆网络 LSTM [11]以及门控循环单元 GRU [12]等模型。Zhang 等人[13]使用 RNN 对云工作负载进行预测，并使用 CPU 和 RAM 指标来评估该方法的性能。然而，RNN 因其递归的网络结构存在梯度消失和梯度爆炸的问题，LSTM 作为其改进模型，通过门控机制能在长序列中保持较稳定的梯度传播。Song 等人[14]使用 LSTM 来学习长期依赖关系，以预测未来连续时间间隔的平均负载和实际负载。Yazdanian 等人[15]提出了一种集成 GAN/LSTM 深度学习架构来预测工作负载网络，该方法使用混合 E2LG 将原始序列分解为具有不同频带的组成成分，采用堆叠 LSTM 块作为生成器，1D-ConvNets 作为判别器，针对每个成分进行单独学习和预测，有效地利用云工作负载时间序列的长期非线性依赖关系。相对于 LSTM，GUR 减少了门控数量，提高了计算效率。Xu 等人[16]针对云计算中资源调度效率低的问题，提出了一种高效的监督学习深度神经网络方法用于云工作负载预测。通过引入滑动窗口机制转换多变量时间序列并结合改进的 GRU 模型，有效地捕捉了工作负载的长期变化特征。RNN 及其变体依赖于递归结构更适合短期云工作负载的预测，基于 Encoder-Decoder 架构的 Transformer [3]采用注意力机制，能够对整个序列的所有位置进行全局的依赖建模，相较来说更适合处理长期时间序列预测任务。Arbat 等[17]提出了一种结合 Transformer 与改进型 Wasserstein-GAN 的云负载预测模型 WGAN-gp Transformer，采用 Transformer 作为生成器，多层感知器作为判别器，用于提升云工作负载预测的精度与推理效率。Zhao 等[18]提出了一种用于准确长期云工作负载预测的多尺度卷积网络 MSCNet 模型，通过多尺度建模原始负载数据，结合 Multi-Scale Patch Block、Transformer 编码器和多尺度卷积块，实现对短期波动与长期趋势的联合学习，从而能够识别云负载中的不同周期模式和长期变化规律。

上述文献虽然将云计算任务时间序列分解为季节性、趋势性以及随机性分量，并使用不同的模型进行预测，最后将各成分的预测值组合起来，一定程度上提高了预测精度，但为每个分解后的序列单独训练一个模型增加了计算资源的消耗，且仍存在误差累积的问题。Lin 等[19]提出的 CycleNet 利用可学习的循环周期对序列中的周期模式进行显示建模，并在残差分量上进行预测，不仅提高了预测精度，还通过参数减少显著提升了计算效率。Liang 等[20]提出了一种周期注意力机制(Periodformer)通过显示建模长期子序列中的周期性，在保证线性计算复杂度的同时显著提升了序列建模能力。受到上述文献的理论启发，本文提出了一种关注周期、趋势、波动的云计算任务数量预测方法，其本质上也是对云计算任务时间序列进行分解，但与现有方法相比，通过显式地建模其固有的周期模式，只使用一个模型对趋势-波动分量进行预测，可减少预测过程中误差的累积，提高预测的准确率。

3. 云平台用户任务的相关概念与问题描述

3.1. 云计算环境建模

云数据中心是云计算体系中最核心的底层基础设施，主要用于集中部署计算服务器、存储设备、网络设备、安全防护设施等硬件设施，以实现对大规模数据的统一存储、高效处理、快速传输、容灾备份

及智能化管理。作为承载云计算服务的物理基础，数据中心需具备高度的稳定性、可扩展性和资源调度能力。

在云数据中心内部，计算资源通常通过集群进行组织和管理。集群由多个通过高速网络互联的计算节点(Node)构成，节点间协同共享资源，并通过调度系统共同执行任务。为满足多样化的业务需求，云平台部署了高性能计算集群、高可用性集群和负载均衡集群等多种类型。各类集群根据任务特性配置不同规格的计算节点，以支持从轻量级请求到大规模并发计算的多种应用场景。

定义 1：云计算节点。在集群中，单台具备独立计算能力与资源管理能力的物理服务器称为计算节点。设一个集群由 N 个计算节点组成，可表示为集合 $G=\{g_1,\cdots,g_i,\cdots,g_N\}$，其中 g_i 表示为集群中第 i 个计算节点。每个计算节点的可用资源可建模为三元组 $g_i=\{gpu_i,cpu_i,mem_i\}$，其中 gpu_i 描述了节点 g_i 的 GPU 资源，可进一步表示为 $gpu_i=\{gpu\_type,gpu\_num\}$，分别为 GPU 的类型和核心数；$cpu_i$ 描述了节点 g_i 的 CPU 处理器，可进一步表示为 $cpu_i=\{cpu\_speed,gpu\_num\}$，分别为 CPU 的计算速度和核心数属性；$mem_i$ 描述了节点 g_i 的内存，可进一步表示为 $mem_i=\{capacity,rspeed,wspeed\}$，分别为内存大小、内存的读取速度和写入速度。

定义 2：单个云计算任务。云计算任务是云平台中可被独立调度、执行和监控的最小计算单元，具有明确的资源需求、执行时长以及生命周期，由云平台调度器分配至特定的虚拟机实例或容器中运行，可形式化地表示为

$$task_i=\{gpu\_type,plan\_gpu,plan\_cpu,plan\_mem,inst\_num,start\_time,end\_time,status\}$$

其中 $gpu\_type,plan\_gpu$ 表示任务对 GPU 资源的需求，包括所需 GPU 型号(如 P100、T4 等)和数量；$plan\_cpu$ 表示任务对 CPU 资源的需求，所需 CPU 核心数；$plan\_mem$ 表示任务对内存资源的需求(GB)；$inst\_num$ 表示任务实际启动的实例数；$start\_time,end\_time$ 表示任务启动时间和完成时间；$status$ 表示任务的状态。

定义 3：云平台任务数量时间序列。在云平台中，用户提交的作业持续动态到达，为刻画云平台负载在时间维度上的变化规律，本文将云计算任务数量定义为一个离散时间序列变量，可形式化表示为

$$X_{t-h+1:t}=\{x_{t-h+1},\cdots,x_i,\cdots,x_t\}$$

其中，t 为当前时间窗口索引，表示目标预测时刻，h 为时间序列的历史长度，即从当前时间 t 向前追溯 h 个历史窗口的数据；x_i 为第 i 个时间窗口内，整体用户提交至云平台并进入调度队列等待执行的计算任务总量，这些任务可能来源于多个用户的不同作业，且其在执行顺序、资源消耗和完成时间上存在差异。

3.2. 云任务数量周期模式存在的客观性

在云计算平台中，任务的不断提交与执行使得其数量在时间维度上呈现出动态变化的特点，往往具有波动性、持续性和周期性等特点。波动性是指由突发或临时的用户请求等随机事件引起的短期波动性负载；持续性是指在较长时间内受到用户规模扩大、业务增长等多种因素影响的整体上升的趋势；周期性是指在一定时间间隔内呈现出规律性的起伏变化。本小节详将细阐述云计算任务周期性特征存在的客观性。

用户请求使用云计算服务，其行为因人类活动规律、业务运行等因素呈现出显著的周期性，例如应遵循自然昼夜规律，大多数用户在白天提交使用服务任务，而晚上提交的频率较低；或因企业策略和运营制度，大型电商平台在大促销期间业务量激增，会需要更多的云资源以维持其运行。周期性不仅在客观上存在，通过历史数据也可观察到任务提交数量在每日存在显著的波峰、低谷波动，如图 1 所示。

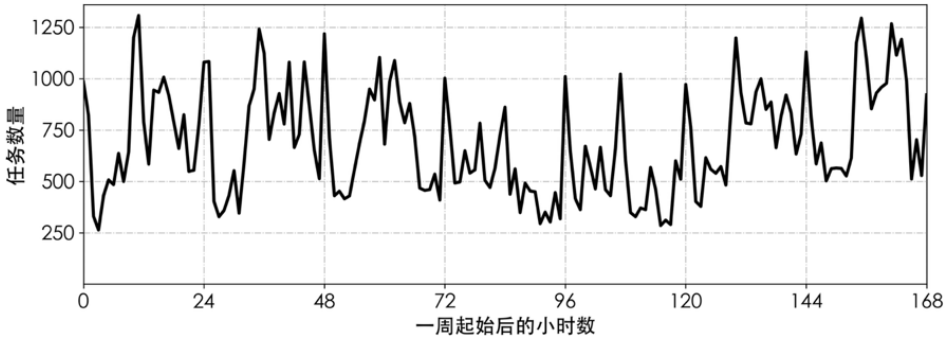

Figure 1. Number of tasks submitted in one week
图 1. 一周内用户提交的任务数量分布

在经典的时间序列分解模型中，时间序列 $X_{t-h+1:t}$ 可以分解为季节性成分、趋势成分和残差成分，即：

$$X_{t-h+1:t} = S_{t-h+1:t} + T_{t-h+1:t} + R_{t-h+1:t}$$

其中，$S_{t-h+1:t}$ 代表季节性分量，表示数据中重复出现的模式或周期；$T_{t-h+1:t}$ 代表趋势分量，表示数据的长期进展或方向；$R_{t-h+1:t}$ 代表残差分量，表示数据中剩余的噪声或随机波动，包括异常值或离群值。通常情况下，与季节性成分和趋势成分相比，残差成分的量级相对较小，因此本文将残差成分 $R_{t-h+1:t}$ 并入趋势成分 $T_{t-h+1:t}$ 中，将时间序列分解为

$$X_{t-h+1:t} = P_{t-h+1:t} + TF_{t-h+1:t} \tag{1}$$

其中，$P_{t-h+1:t}$ 表示固有的周期分量(periodic components)，$TF_{t-h+1:t}$ 表示趋势－波动分量(Trend-Fluctuation components)。

本文将云计算任务数量时间序列 $X_{t-h+1:t}$，分解为周期分量 $P_{t-h+1:t}$ 和趋势－波动分量 $TF_{t-h+1:t}$，其中周期分量 $P_{t-h+1:t}$ 由单个周期模式 Q 循环建模获得。

3.3. 问题描述

本文的研究动机是对用户任务到达数量进行精确预测，即根据历史任务记录为云平台预测一段时间内即将来到的云计算任务请求数量，从而及时为用户分配资源以保证 QoS。形式上可描述为：对某一云数据中心的某一个集群，收集其在一段时间内用户提交的任务，并统计生成关于过去 h 个时间步到达云平台的云计算任务数量时间序列 $X_{t-h+1:t}$，构建预测模型 $f(\cdot)$ 后，获得到未来 n 个时间步的达到任务数量

$$Y_{t+1:t+n} = f(X_t) = \{y_{t+1}, \cdots, y_i, \cdots, y_{t+n}\} \tag{2}$$

其中 y_i 表示未来第 i 时刻预计到达的任务数量。

4. 周期模式驱动的云任务数量预测方法

4.1. 融合周期、趋势、波动因素的预测模型整体框架

为了能准确预测云计算任务数量，并且减少长期时间序列预测中误差累积的问题，本文提出了一种关注周期、趋势、波动特征的预测模型(P-T-F Model)，通过显式建模云平台任务数量时间序列中的周期分量，对分解出的非周期成分进行建模和预测。整体流程如图 2 所示，主要包括以下步骤：

(1) 数据预处理：从云平台原始任务日志中，构建任务数量时间序列，并进行异常值处理与标准化；

(2) 周期分量建模：利用周期注意力机制，提取具体的周期性模式特征，获得一个可学习的全局循环周期表示；

(3) 趋势－波动建模：将时间序列中去除周期后的非周期部分输入 Transformer 模型，捕捉时间序列中长期趋势与随机波动部分；

(4) 分量结果融合：将预测得到的未来周期分量与趋势－波动分量相加，获得最终的任务数量预测结果。

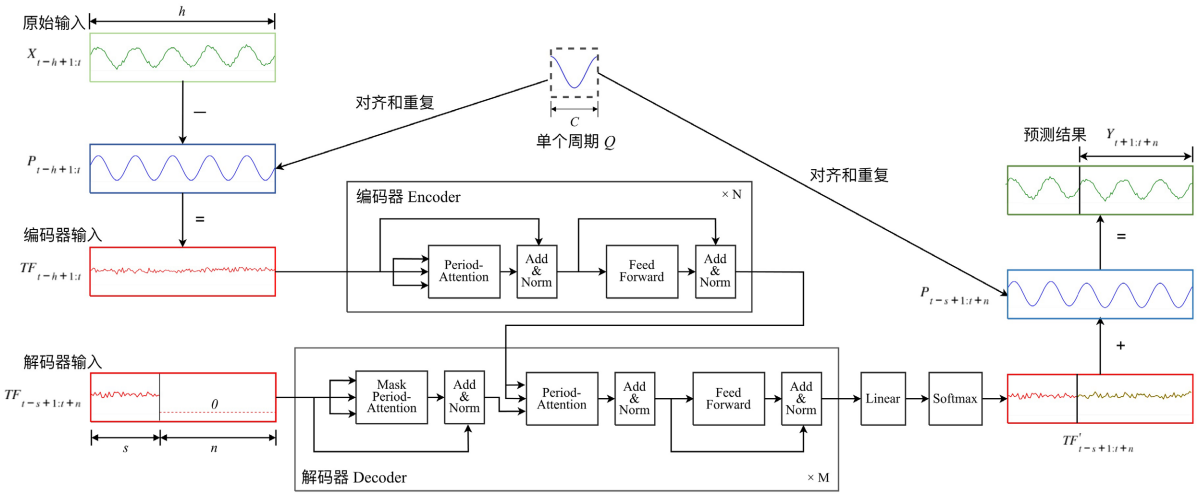

Figure 2. Overall framework of P-T-F model
图 2. 云任务数量预测模型整体框架

4.2. 周期建模与周期分量生成

在云环境中，任务的达到模式通常呈现出稳定的周期性循环结构(如每日循环或每周循环)，在时间序列中表现出显著且明确的周期模式，针对云计算任务数量时间序列中的周期分量，本小节进行显式建模以生成单个可学习的周期分量。

周期长度 C 定义为时间序列中某一稳定重复模式的最小完整单位长度，其准确设定有助于捕捉真实的周期结构。本文使用自相关函数(autocorrelation functions, ACF)来确定周期长度 C。自相关函数可衡量时间序列与其滞后值之间的相关性，其数学表达式为：

$$\mathrm{ACF}=\frac{\sum_{t=1}^{N-k}\left(x_t-\bar{x}\right)\left(x_{t+k}-\bar{x}\right)}{\sum_{t=1}^{N}\left(x_t-\bar{x}\right)^2} \tag{3}$$

其中 N 表示观测值的总数，x_t 表示时间 t 时时间序列的值，k 为滞后时间，$\bar{x}$ 为时间序列值的均值。当滞后时间 k 与数据的周期对齐时，ACF 值会显著上升，最大的峰值对应于与数据集中最大周期长度对齐的滞后时间；反之，若数据缺乏周期性，则不会观察到显著的峰值或谷值。

针对已确定单个可循环周期长度的时间序列，将单个可循环的周期 $Q\in\mathbb{R}^{C\times D}$ 初始化为零矩阵，其中 C 为周期长度，D 为输入时间序列的维度，本文中 $D=1$。周期 Q 作为可学习的参数矩阵，与预测模型一起训练，直接参与模型的前向传播和反向传播，其参数会根据损失函数的梯度进行调整，从而逐渐学习到数据中的周期性模式。周期 Q 在通道之间是全局共享的，通过执行对齐和重复操作可得到与原始时间序列 $X_{t-h+1:t}$ 具有相同长度的周期性分量 $P_{t-h+1:t}$。

4.3. 利用周期注意力机制完善周期分量

准确提取时间序列中周期成分是本模型提升长期时间序列预测精度的关键，因此为进一步增强周期

模式的表达能力，本文提出了一种周期注意力机制对周期 Q 的学习进行引导，在时间序列内部按照周期长度 C 划分为多个子序列，在对应位置间建立显式的连接以实现周期信息的聚合。

标准的多头注意力机制通过计算所有时间点间的交互来捕获全局依赖关系，但其注意力分布可能过于分散，且在处理长期时间序列时易受到非周期性成分的噪声干扰。于理论而言，该机制能够学习到时间序列中的周期模式，但这种学习是隐式的，模型需要从大量数据中自行推断周期成分的存在及其长度，在数据不足或噪声明显时，难以保证模型对周期成分的准确建模。

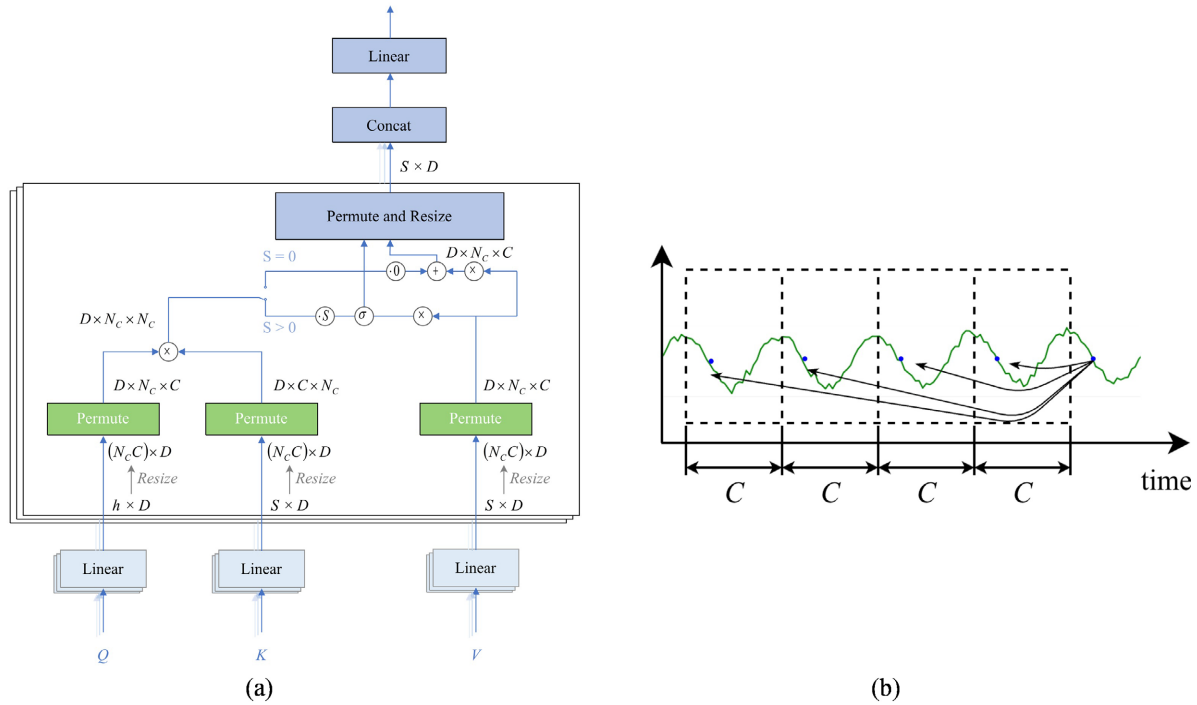

Figure 3. The periodic attention mechanism
图 3. 周期注意力机制

周期注意力机制的具体结构如图 3(a)所示。输入首先经过线性变换得到查询向量(Query, Q)、键向量(Key, K)、值向量(Value, V)，随后根据 4.2 小节确定的周期长度 C 对 Q、K、V 的序列维度进行重构，Q、K、V 的形状 (h,d) 可重新排列为 (D,N_C,C)。这一重构操作在物理意义上将连续的长时间序列划分为 N_C 个子序列，每个子序列内聚合了一个周期的信息，包含 C 个时间点，如图 3(b)所示。在此基础上，周期注意力机制的注意力计算在 N_C 这个维度上进行，对于每个当前周期子序列内的每一个特定时间点，其查询向量(Q)会与所有周期子序列内的同一相位时间点的键向量(K)计算注意力分数，从而能够有效地捕获稳定且可重复的周期模式。另外，为防止注意力机制过拟合噪声，利用缩放因子 s 来调节注意力分数对其输出的影响，周期注意力计算具体可表示为

$$\text{Attention}(Q,K,V)=\begin{cases}\sigma\left(\dfrac{QK^{\mathrm{T}}\cdot s}{\sqrt{C}}\right)V, & s>0\\ \delta(V), & s=0\end{cases} \tag{4}$$

其中，σ 是 Softmax 函数，δ 是一个激活函数。当 $s>0$ 时，注意力分数 QK^{T} 乘以 s 以控制其对 V 的影响，再与 V 相乘获得最终的注意力分数；当 $s=0$ 时，注意力机制被取消，并被一个非线性激活操作 δ 替

代。最后，输出会重新排列并调整为与原始输入相同的形状，最终通过拼接和线性变换生成周期注意力子层的输出。

4.4. 基于 Transformer 模型的趋势－波动分量预测

周期成分建模后，时间序列中仍存在无法由周期结构解释的趋势变化与随机波动的部分，这部分信息对预测结果同样重要。为此，本文通过 Transformer 模型对剩余的趋势－波动项进行建模与预测。

4.4.1. 趋势－波动原始分量的获取

从原始时间序列 $X_{t-h+1:t}$ 中移除周期成分，获得关于趋势－波动部分的时间序列 $TF_{t-h+1:t}$ 作为编码器的输入：

$$TF_{t-h+1:t} = X_{t-h+1:t} - P_{t-h+1:t} = \left\{tf_{t-h+1}, \cdots, tf_i, \cdots, tf_t\right\} \tag{5}$$

其中，f_i 表示第 i 时刻的趋势－波动项。由于无法直接获得 $P_{t-h+1:t}$，需要对由 4.2 小节构建的单个周期项 Q 进行适当的对齐和重复得到长度为 h 的序列，具体为

$$P_{t-h+1:t} = \left\{ \overbrace{Q, \cdots, Q}^{\left\lfloor \frac{h}{C} \right\rfloor \times C}, \overbrace{Q(t)_{0:t \bmod C}}^{h \bmod C} \right\} \tag{6}$$

其中，将 Q 左移 $t \bmod C$ 个位置得到 $Q(t)_{0:t \bmod C}$，$t \bmod C$ 为当前时间步 t 在单个周期中相对位置索引，即 h 与周期对齐后的偏移量；再将单个周期 Q 重复 $N_C = \left\lfloor \frac{h}{C} \right\rfloor$ 次与 $Q(t)$ 连接形成完整的周期成分。

4.4.2. 趋势－波动特征提取的编码器

编码器 Encoder 由 N 个相同的层堆叠而成，逐层提取输入序列中的深层特征表示，每一层的输出作为下一层的输入，其关系可表示为 $X^l = \text{Encoder}\left(X^{l-1}\right)$。每层结构由周期注意力机制(Period-Attention)与前馈神经网络 FFN 两个子层组成，用于捕捉输入序列中的周期性成分和特征表示，其中周期注意力机制已在 4.3 小节中详细描述。

前馈神经网络通过两次线性变换和一次非线性激活函数映射，能够进一步丰富输入特征的表示，具体计算如下：

$$\text{FFN}(x) = \max\left(0, xW_1 + b_1\right)W_2 + b_2 \tag{7}$$

其中，W_1 和 W_2 是学习得到的权重矩阵，b_1 和 b_2 是偏置项。通过这种非线性映射，FFN 能够学习到更复杂的模式和特征，使得模型具有更强的拟合能力。

在每个子层后使用残差连接，将输入与输出相加，保留输入信息，可缓解梯度消失或爆炸的问题。残差连接后应用层归一化，规范数据特征分布，可减少偏差，提升模型泛化能力。

4.4.3. 趋势－波动分量预测的解码器及输出

本文采用生成式解码器一步生成目标长度的预测结果，减少误差在时间维度上的递归累积。从 $TF_{t-h+1:t}$ 中提取最近 k 个周期的子序列 $TF_{t-s+1:t}$ 作为解码器输入的初始化序列，由于解码器通过交叉注意力(cross-attention)机制与编码器的输出交互进行预测，而交叉注意力机制要求 Value 的长度必须大于 Key 和 Query 的长度，因此需要使用占位符填充解码器输入使得预测序列在长度上对齐，因此使用长度为n的占位符 0 填充以匹配最终的预测长度，解码器输入为

$$TF_{t-s+1:t+n}=\left\{\overbrace{tf_{t-s+1},tf_{t-s+2},\cdots,tf_t}^{s=k\times C},\overbrace{0,\cdots,0}^{n}\right\} \tag{8}$$

与编码器类似，解码器同样采用多层结构，由 M 个相同的层堆叠而成。任意两个相邻的解码器层之间的关系可表示为 $X^l=\mathrm{Dncoder}\left(X^{l-1}\right)$，每一层由掩码周期注意力机制(Masked Period-Attention)、交叉周期注意力机制(Cross-Period-Attention)和前馈神经网络三个子层组成，其中第一个周期注意力子层采用掩码(Masked)操作，确保了解码输出仅依赖于时间步 t 之前的输出，而无法获取 t 之后的输出信息；第二个交叉周期注意力子层的键和值均来自编码器的输出，使解码器能够动态地从编码器的输出中提取相关信息。

4.5. 最终预测结果的生成

完成周期分量建模与趋势－波动成分预测后，最终的多步预测结果由这两个分量融合得到。解码器输出的趋势－波动分量预测值 $TF'_{t-s+1:t+n}$ 与周期性成分 $P_{t-s+1:t+n}$ 相加，得到最终预测值：

$$Y_{t-s+1:t+n}=TF'_{t-s+1:t+n}+P_{t-s+1:t+n} \tag{9}$$

其中，周期成分 $P_{t-s+1:t+n}$ 通过单个周期Q进行对齐和重复得到长度为 $s+n$ 的序列：

$$P_{t-s+1:t+n}=\left\{\overbrace{Q,\cdots,Q}^{\left\lfloor\frac{s+n}{C}\right\rfloor\times C},\overbrace{Q(t)_{0:(t+n)\bmod C}}^{(s+n)\bmod C}\right\}$$

最后，从 $t+1$ 时刻开始选取n个时间步的预测结果，形成完整的未来任务数量序列 $Y_{t+1:t+n}$。

5. 实验与结论

5.1. 实验数据来源

为了验证本文提出方法的有效性，我们采用 Alibaba-cluster-gpu-v2020 公开数据集[21]进行实验验证。该数据集来源于阿里巴巴人工智能平台，记录了一个包含 6500 余个 GPU (大约 1800 台机器)的大型生产集群在连续两个月内的实际工作负载情况。数据集内容涵盖不同层次工作负载(如作业、任务、实例)的到达时间、调度时间、完成时间，以及在 GPU、CPU、内存和磁盘的等资源上的请求与实际使用情况，同时还记录了机器的硬件规格与其资源利用率。具体数据来源是 pai_task_table.cvs 文件，该文件内记录了每个任务的核心信息，包括任务的名称、实例数量、任务状态、启动与完成时间、申请的计算与存储资源。为便于时间序列建模，本文对所有任务按照达到时间进行统计，并以小时为粒度形成云计算任务达到数量时间序列，为后续预测模型构建提供输入基础。

5.2. 评价指标

模型训练阶段使用均方误差(Mean Squared Error, MSE)作为损失函数，模型性能评估阶段使用 MSE 和平均绝对误差(Mean Absolute Error, MAE)作为评价指标。两项指标均反应模型预测值与真实值之间的误差，MAE 和 MSE 值越小表示模型的预测误差越小，模型的预测能力越强。其计算公式如下：

$$\mathrm{MAE}=\frac{1}{n}\sum_{i=1}^{n}\left|y_i-\hat{y}_i\right| \tag{10}$$

$$\mathrm{MSE}=\frac{1}{n}\sum_{i=1}^{n}\left(y_i-\hat{y}_i\right)^2 \tag{11}$$

其中， y_i 与 $\hat{y}_i$ 分别为真实值与预测值，n 为样本数量。

5.3. 实验过程与结果

5.3.1. 实验过程

本文基于 PyTorch 框架编程实现了 P-T-F 模型，并在单张显存 24 GB 的 NVIDIA RTX 4090 GPU 上完成全部训练与测试。实验整体流程主要包括数据预处理、特征构建、模型训练三个阶段：

(1) 数据预处理阶段：首先对 pai_task_table.csv 文件进行数据清洗与结构规整，剔除无效任务记录及缺失字段。对任务的达到时间与结束时间进行统一时间戳标准化处理，并按小时粒度统计任务到达数量，形成时间序列数据。为消除不同时间段内任务规模差异所带来的数值影响，提高模型训练稳定性，对时间序列进行标准化处理。具体而言，采用 StandardScaler 对任务数量进行 Z-score 标准化，基于训练数据计算均值与标准差，并据此对训练集、验证集和测试集进行一致的标准化变换。

(2) 特征构建阶段：在原始任务数量时间序列中分解周期成分与趋势–波动成分，并通过滑动窗口机制构建输入样本。设定时间窗口长度为 $h=96$，预测步长为 $l=24,48,96$，单个周期长度为 $C=24$，则模型的输入为过去 h 小时的任务数量序列，输出为未来 l 小时的任务数量预测值。

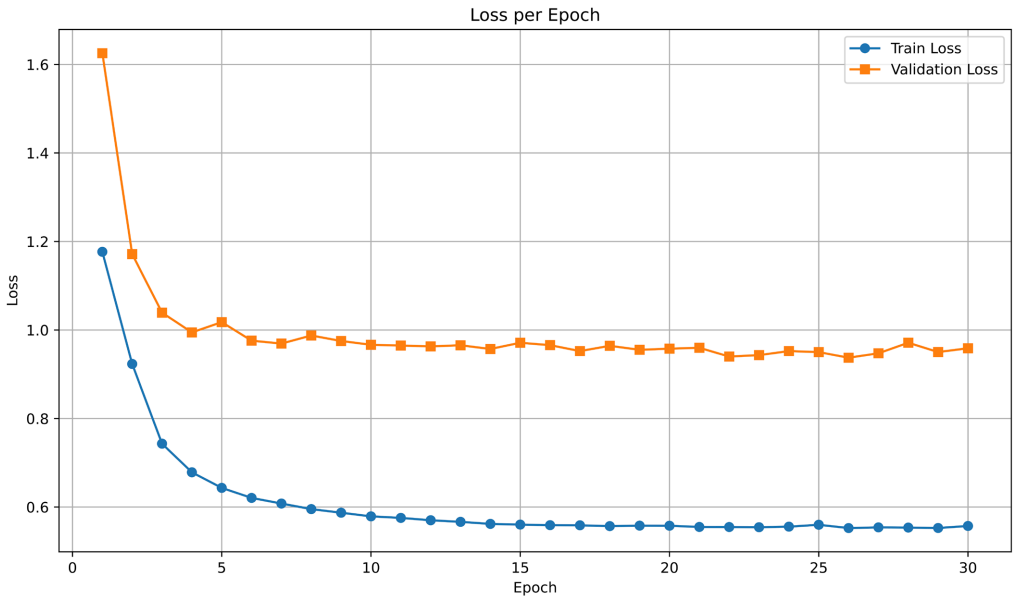

Figure 4. Loss curves of training and validation sets
图 4. 训练集与验证集 Loss 曲线

(3) 模型训练阶段：模型的损失函数使用 MSE，优化器选用 Adam，初始学习率设置为 1×10^{-4}，批次大小设为 64，最大训练轮数为 30。为抑制过拟合并提高泛化能力，训练过程中引入早停机制(patience = 5)，以验证集 MSE 作为监控指标，连续 5 个 epoch 未取得精度提升则终止训练。图 4 展示了训练集与验证集 Loss 随 epoch 变化的曲线，训练 Loss 在前 8 轮快速下降，随后进入平稳收敛区间，验证 Loss 在第 15 轮达到最低值 0.95 并保持平台震荡，未出现明显回升，表明模型已充分收敛且无过拟合风险。总体而言，Loss 曲线平滑、无尖刺，说明学习率、批次大小等参数设置合理，训练过程稳定。

5.3.2. 周期长度设定与周期模式分析

本文利用 ACF 分析确定云平台任务数量时间序列的周期长度 C，其 ACF 结果如图 5 所示。在图中

可以观察到，低阶滞后($Lag < 10$)自相关显著偏高，随后快速衰减，说明云平台任务达到数量序列具有短期相关性与惯性，符合云平台工作负载“连续达到”的基本特征。在滞后阶数 $Lag \approx 24$ 附近出现显著的局部峰值，且在其整数倍处均出现显著的自相关峰值，表明云平台任务达到模式具有稳定且显著的日周期特征，这一结果与 3.2 小节关于周期性存在的客观分析相一致。随着滞后阶数的增加，自相关系数略有降低，但周期性峰值仍然清晰可辨，说明该周期模式在较长时间尺度上都保持稳定，适用于本文基于周期建模的长期时间序列预测方法。基于上述 ACF 结果，可初步判断云平台任务达到数量序列存在以 24 为周期的稳定循环结构。然而，由于周期长度 C 作为周期建模中的关键超参数，其取值可能直接影响模型对周期模式的刻画能力，仅依赖统计分析仍不足以全面评估其合理性。

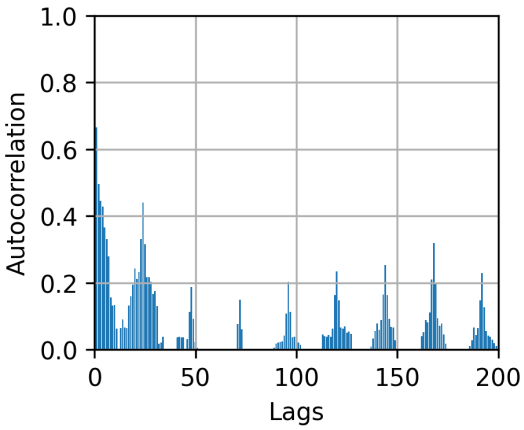

Figure 5. Visualization of ACF results
图 5. ACF 结果可视化

Table 1. Impact of different cycle lengths C on prediction performance
表 1. 不同周期长度 C 对模型预测性能的影响

周期长度	指标	$l = 24$	$l = 48$	$l = 96$
$C = 12$	MSE	0.694	0.852	1.065
$C = 24$	MSE	**0.625**	**0.758**	**0.952**
$C = 36$	MSE	0.651	0.791	0.998
$C = 48$	MSE	0.702	0.865	1.086

因此，为进一步验证周期长度 C 选择的合理性，本小节围绕该关键超参数开展了敏感性分析实验。在保持模型结构与训练参数不变的条件下，分别选取 $C = 12, 24, 36, 48$ 等具有代表性的取值进行对比实验。实验结果如表 1 所示，当 C 取值偏离真实周期时，模型预测误差明显上升。周期过短($C = 12$)导致模型无法捕捉完整的周期模式；周期过长($C = 48$)则将多个真实周期合并至同一窗口中，引入了相位错乱和冗余噪声，削弱周期注意力对关键时间位置的辨识能力；而当 C 取值位于真实周期附近时($C = 36$)，模型整体预测精度与最佳周期长度($C = 24$)更为接近，模型捕捉到的周期模式仍能完整覆盖一个真实周期并保留一定冗余，使模型有机会捕捉到云平台任务达到序列中真实的周期结构。综合 ACF 分析结果、云平台任务达到的实际物理周期意义以及敏感性实验结论，本文将周期长度 C 设置为 24。

为进一步增强模型的可解释性，本小节对模型从历史数据中学习得到的周期模式进行了可视化，如图 6(a)所示，并与图 6(b)所示的部分原始任务数量变化情况进行对比。可以观察到，模型所学习到的周期

模式在结构上与真实任务数量的分布高度一致，说明模型能够有效识别到周期内不同时间段任务数量的差异，并在预测过程中对关键时间位置赋予更高关注度。此外，相较于原始序列中存在的随机波动，模型学习到的周期模式表现出更加平滑且稳定的峰谷结构，说明所提出方法更侧重于捕捉跨周期重复出现的稳定周期特征，而非对短期噪声进行过拟合。

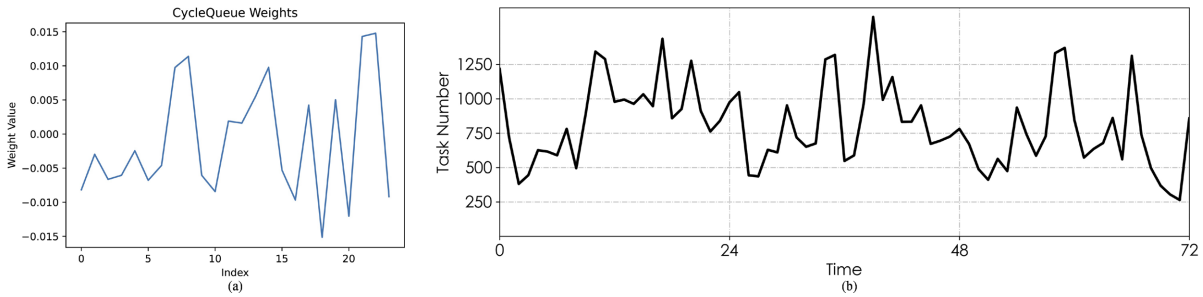

Figure 6. Comparison of learned daily periodic pattern and original task distribution
图 6. 日周期模式学习结果及其原始任务分布对比

5.3.3. 预测结果评估与案例分析

为从整体层面评估 P-T-F 模型的预测性能，本小节首先给出了测试集上真实任务数量与预测结果的整体对比曲线，如图 7 所示。可以观察到，预测序列在全时间范围内能够较好地跟随真实任务数量的变化趋势，在多数时间段内能够与真实曲线保持较高的一致性，未出现明显的趋势漂移，说明 P-T-F 模型在长期预测场景下具备良好的稳定性与一致性。

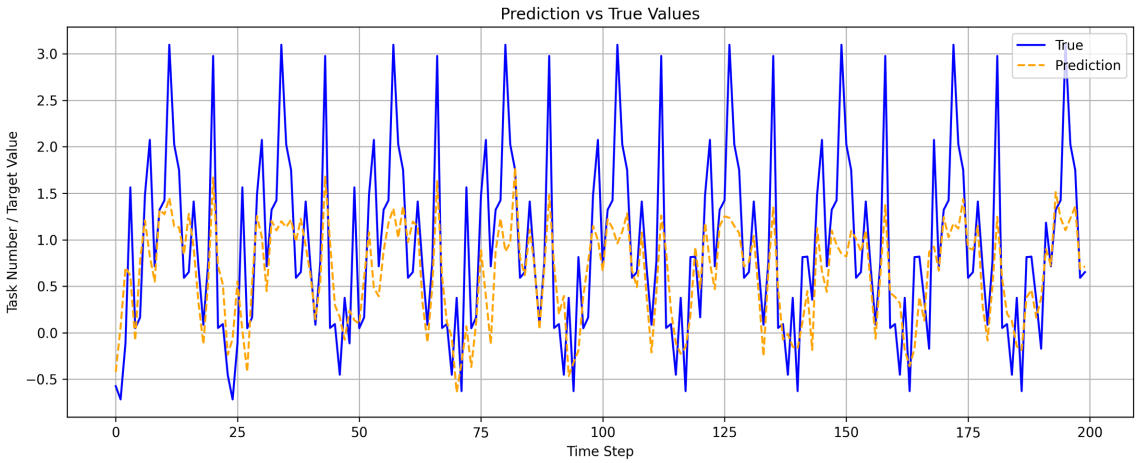

Figure 7. True values vs. predicted values
图 7. 真实值与预测值对比可视化

为进一步评估模型在不同预测步长下的性能表现，本小节展示了在相同历史输入条件下，预测步长分别为 24、48 和 72 的代表性预测结果，如图 8 所示，其中黑色竖线表示预测区间的起始位置。从预测结果可以看出，在预测步长为 24 时，模型预测曲线在幅值和相位上均与真实序列保持较高一致性，能够较准确地刻画周期性波动及其局部峰谷结构。随着预测步长增加至 48 和 72，预测结果整体趋势仍与真实序列保持一致，但在局部剧烈波动区域，预测曲线与真实值仍有较大误差，对突发尖峰的刻画能力稍有不足。这表明在一步生成较长预测序列时，模型更倾向于优先保持整体周期结构与长期趋势的稳定性。

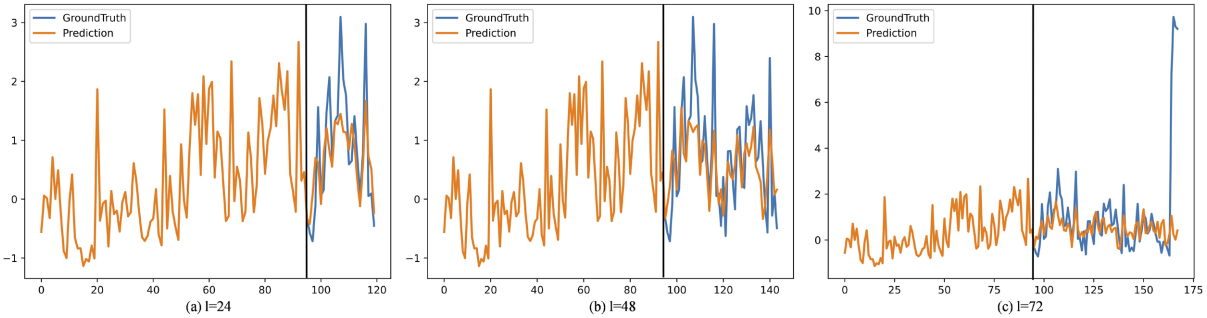

Figure 8. True values vs. Predictions at forecast horizons of 24, 48, and 72
图 8. 预测步长为 24、48、72 的真实值与预测值结果对比

综上分析，P-T-F 模型在周期主导的稳定负载场景中表现出较强的预测能力，预测曲线能够准确跟随真实序列的周期起伏变化，峰谷位置与变化节律保持良好一致，表明模型在周期特征建模和长期趋势保持方面具有显著优势，适用于云平台长期任务数量预测。然而，在极端突发负载或快速模式转变场景下，模型对非周期性尖峰的刻画仍存在一定局限。

5.4. 对比实验

本文将 ARIMA、LSTM、Transformer 作为基准模型，通过对比试验验证 P-T-F 模型性能优势，所有模型均在相同数据集与相同数据划分下进行训练与测试。基准模型设置如下。

ARIMA 模型：作为经典时间序列预测基准，采用 ARIMA 模型对平稳化后的差分序列进行建模。模型的最优超参数(p,d,q)通过 Akaike 信息准则(AIC)进行自动搜寻与确定，最终选定参数为$(p,d,q)=(2,1,2)$，适用于平稳化后的差分序列。

LSTM 模型：作为深度学习中处理序列数据的经典模型，本小节构建了一个两层堆叠的 LSTM 网络。每层隐藏层单元数均为 28，使用 Tanh 作为循环体内的激活函数，模型通过 Adam 优化器进行训练，初始学习率设置为 1×10^{-3}。

Transformer 模型：本小节复现了 Vaswani 等人提出的标准 Encoder-Decoder 架构，包含 6 个编码器层与 6 个解码器层，模型隐藏层的维度设置为 256，采用多头自注意力机制($h=8$)。该模型是评估本文提出的周期注意力机制能否在强大基线之上带来显著性能提升的关键参照。

实验结果如表 2 所示，可以发现：传统的统计方法 ARIMA 难以处理云任务数量序列中的非线性与多周期特征；基于深度学习的序列模型 LSTM、Transformer 在短期预测上表现较优，但未充分利用任务数量序列的周期性信息；本文提出的 P-T-F 模型相较于以上对比模型取得了稳定且一致的性能提升，预测结果更加平滑且稳定，误差波动范围较小。综上所述，本文提出的模型在长期任务数量预测中具有更高的预测精度与更强的周期适应能力，为云平台的资源规划与能耗优化提供了可行的技术支撑。

Table 2. Prediction results of different models
表 2. 对比实验结果

模型	指标	$l=24$	$l=48$	$l=96$
ARIMA	MSE	0.985	1.225	1.580
	MAE	0.782	0.945	1.132
LSTM	MSE	0.725	0.890	1.150
	MAE	0.610	0.735	0.890

续表

Transformer	MSE	0.685	0.825	1.050
	MAE	0.585	0.695	0.835
P-T-F Model	MSE	**0.625**	**0.758**	**0.952**
	MAE	**0.542**	**0.648**	**0.775**

6. 结束语

云平台任务数量具有周期性显著、波动性强等特点，且长期时间序列预测存在的误差易累积问题，传统方法难以在复杂负载环境下实现稳定有效的预测。为此，本文提出了一种关注周期、趋势、波动的预测方法，通过可学习的周期成分建模与周期注意力机制，有效提升了周期模式提取效果；生成式Transformer解码结构减缓了长期预测中的误差扩散。实验结果表明，所提出的方法在长期预测精度、稳定性及周期特征刻画能力方面均显著优于多种基准模型，能够更有效地支持云平台资源规划与调度决策。

尽管本文方法在复杂云负载预测场景中表现出较好的预测精度与稳定性，但仍存在一定局限性，模型对周期长度的设定具有较强依赖，在周期发生变化或呈现多尺度特征的场景下，其建模灵活性仍有待提升。未来的工作将从两个方面展开：第一，进一步研究自适应周期检测与多周期融合建模方法，以增强模型对非平稳负载和复杂周期模式的刻画能力；第二，将负载预测模型与云计算任务资源调度策略相结合，探索面向能耗与资源利用率等多目标优化的任务调度机制，从而充分发挥预测结果在实际云资源管理与调度决策中的指导作用。

参考文献

[1] 中国信息通信研究院. 云计算白皮书[R]. 北京: 中国信息通信研究院, 2024.

[2] Feng, B. and Ding, Z. (2025) Application-Oriented Cloud Workload Prediction: A Survey and New Perspectives. *Tsinghua Science and Technology*, **30**, 34-54. https://doi.org/10.26599/tst.2024.9010024

[3] Vaswani, A., Shazeer, N., Parmar, N., *et al*. (2017) Attention Is All You Need. *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*, Long Beach, 4-9 December 2017, 6000-6010.

[4] Calheiros, R.N., Masoumi, E., Ranjan, R. and Buyya, R. (2015) Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications' QoS. *IEEE Transactions on Cloud Computing*, **3**, 449-458. https://doi.org/10.1109/tcc.2014.2350475

[5] Bi, J., Zhang, L., Yuan, H. and Zhou, M. (2018) Hybrid Task Prediction Based on Wavelet Decomposition and ARIMA Model in Cloud Data Center. 2018 *IEEE* 15*th International Conference on Networking, Sensing and Control* (*ICNSC*), Zhuhai, 27-29 March 2018, 1-6. https://doi.org/10.1109/icnsc.2018.8361342

[6] Islam, S., Keung, J., Lee, K. and Liu, A. (2012) Empirical Prediction Models for Adaptive Resource Provisioning in the Cloud. *Future Generation Computer Systems*, **28**, 155-162. https://doi.org/10.1016/j.future.2011.05.027

[7] Zhongda, T., Shujiang, L., Yanhong, W. and Yi, S. (2017) A Prediction Method Based on Wavelet Transform and Multiple Models Fusion for Chaotic Time Series. *Chaos, Solitons & Fractals*, **98**, 158-172. https://doi.org/10.1016/j.chaos.2017.03.018

[8] Jeddi, S. and Sharifian, S. (2020) A Hybrid Wavelet Decomposer and GMDH-ELM Ensemble Model for Network Function Virtualization Workload Forecasting in Cloud Computing. *Applied Soft Computing*, **88**, Article ID: 105940. https://doi.org/10.1016/j.asoc.2019.105940

[9] Kumar, J. and Singh, A.K. (2020) Decomposition Based Cloud Resource Demand Prediction Using Extreme Learning Machines. *Journal of Network and Systems Management*, **28**, 1775-1793. https://doi.org/10.1007/s10922-020-09557-6

[10] Feng, B., Ding, Z. and Jiang, C. (2023) FAST: A Forecasting Model with Adaptive Sliding Window and Time Locality Integration for Dynamic Cloud Workloads. *IEEE Transactions on Services Computing*, **16**, 1184-1197. https://doi.org/10.1109/tsc.2022.3156619

[11] Hochreiter, S. and Schmidhuber, J. (1997) Long Short-Term Memory. *Neural Computation*, **9**, 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[12] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., *et al*. (2014) Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *Proceedings of the* 2014 *Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Doha, October 2014, 1724-1734. https://doi.org/10.3115/v1/d14-1179

[13] Zhang, W., Li, B., Zhao, D., Gong, F. and Lu, Q. (2016) Workload Prediction for Cloud Cluster Using a Recurrent Neural Network. 2016 *International Conference on Identification*, *Information and Knowledge in the Internet of Things* (*IIKI*), Beijing, 20-21 October 2016, 104-109. https://doi.org/10.1109/iiki.2016.39

[14] Song, B., Yu, Y., Zhou, Y., Wang, Z. and Du, S. (2017) Host Load Prediction with Long Short-Term Memory in Cloud Computing. *The Journal of Supercomputing*, **74**, 6554-6568. https://doi.org/10.1007/s11227-017-2044-4

[15] Yazdanian, P. and Sharifian, S. (2021) E2LG: A Multiscale Ensemble of LSTM/GAN Deep Learning Architecture for Multistep-Ahead Cloud Workload Prediction. *The Journal of Supercomputing*, **77**, 11052-11082. https://doi.org/10.1007/s11227-021-03723-6

[16] Xu, M., Song, C., Wu, H., Gill, S.S., Ye, K. and Xu, C. (2022) esDNN: Deep Neural Network Based Multivariate Workload Prediction in Cloud Computing Environments. *ACM Transactions on Internet Technology*, **22**, 1-24. https://doi.org/10.1145/3524114

[17] Arbat, S., Jayakumar, V.K., Lee, J., Wang, W. and Kim, I.K. (2022) Wasserstein Adversarial Transformer for Cloud Workload Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, **36**, 12433-12439. https://doi.org/10.1609/aaai.v36i11.21509

[18] Zhao, F., Lin, W., Lin, S., Tang, S. and Li, K. (2025) MSCNet: Multi-Scale Network with Convolutions for Long-Term Cloud Workload Prediction. *IEEE Transactions on Services Computing*, **18**, 969-982. https://doi.org/10.1109/tsc.2025.3536313

[19] Hu, X., Lin, S., Lin, W., Mo, R., Wu, W. and Zhong, H. (2024) CycleNet: Enhancing Time Series Forecasting through Modeling Periodic Patterns. *Proceedings of the* 38*th Annual Conference on Neural Information Processing Systems* (*NeurIPS* 2024), Vancouver, 10-15 December 2024, 106315-106345. https://doi.org/10.52202/079017-3373

[20] Liang, D., Zhang, H., Yuan, D. and Zhang, M. (2024) Periodformer: An Efficient Long-Term Time Series Forecasting Method Based on Periodic Attention. *Knowledge-Based Systems*, **304**, Article ID: 112556. https://doi.org/10.1016/j.knosys.2024.112556

[21] Alibaba (2021) Alibaba Cluster Trace Program: Cluster-Trace-Gpu-v2020. https://github.com/alibaba/clusterdata/tree/master/cluster-trace-gpu-v2020